# TODO:

- include all the assumptions of all the packages (if not specified, set to X; uses algorithm X; etc.)?

# 1. General information

……

Please note: While the following instructions may also be applicable to other operating systems, they were primarily written for Windows users.

## Bridging Java and R

The front-end of the application is developed in Java, with a GUI in Swing. It communicates with the suite of packages of the R Project for Statistical Computing using the JRI (Java-R interface) library.

In order to properly install JRI for the purposes of the tool, you can follow these steps (for Windows):

1. Install R, Java and JRI.
2. Set the value of the environment variable R\_HOME to where your installation of R lies, e.g. C:\Program Files\R\R-3.1.1
3. Add to the value of the environment variable Path the path to R.dll (%R\_HOME%/bin/x64) and to jri.dll (%R\_HOME%/library/rJava/jri/x64)

# 2. System requirements

Since both R and Java are platform-independent, there are no specific requirements on the platform or operating system of the machine the application will run on. However, a proper installation of Java (version 7 and higher), JRI (version ….), R (version 3.1.1 and higher) and all the necessary R packages is required. Please note that it is important to synchronize the architecture versions of Java and R, i.e. make sure both installations are either 32-bit or 64-bit.

For the C executable (implementing iMLP) to work, the cygwin1.dll library needs to be on the Path as well.

The tool requires the following R packages to be installed in the R environment: rJava, forecast, JavaGD, XLConnect, FNN, kknn, nnet, scales, reshape, RSNNS, vars, psych, nortest, tseries, bfast. (The order is of no importance.)

# 3. Running the tool

The tool can be launched from the command line by running “java -jar MLPtoR.jar” or double-clicking the executable .jar file. The following files and folders need to be in the same folder as the executable:

- the “lib” folder, containing javagd.jar, jri.jar, jriengine.jar, jxl.jar and rengine.jar

- the scripts.R file (Note: might be removed in the future and built inside the application.)

- the c.exe

# 4. Loading files

When the application is run, there is not much that can be done unless a file with data is loaded. To load a file, click on File -> Load in the top menu. A dialog window will appear, enabling the user to browse the file system and select a file to open. Currently only Excel spreadsheets are supported, i.e. files with extensions .xls and .xlsx. For this reason, only directories and Excel files are visible in this dialog. The Excel file is expected to be in a format where each time series is in a separate column, optionally with a header specifying its name, and where the first column can, but does not have to contain the timestamps for all time series.

At the side of the dialog, there are several options available. The checkbox “Use the first column as labels for time periods” should be selected if the file the user wishes to load contains the timestamps for the time series in the first column. If it is not selected, the tool will use ordinal numbers as timestamps, i.e. the first entry will have a timestamp equal to 1, the next to 2, etc.

Similarly to the timestamps, the names of the time series can be specified in the file. Select “Use the first row as column names” if the file contains names of the time series; otherwise, it is possible to generate the names in the tool. When the option “No column names in the file, use placeholders” is selected, the tool will set the name of each time series to be “X<number>”, where the “<number>” is unique for each time series. In future versions, it will also be possible to input custom names for the time series in the loading phase, but this option is not available at the moment.

If the file does not meet the criteria selected in the options, an error may occur resulting in the failure to load any files.

Only the first sheet of any file is taken into account; however, this may change in future versions.

After the file is successfully loaded, the data can be seen in the “Data” tab and the names of all time series are propagated to all option panels.

# 5. Plotting

The tool supports several types of plots useful for the time series analysis.

## 5.1. Classical time series plot

When one or several time series are selected in the list in the top-left corner of the “Plot Settings” tab, the “Plot selected time series” button next to the list produces a plot containing all the specified entries. They are all resized to a common scale, so in case the differences in the values of two time series are too big, the combined plot may not be very clear.

The plot can be found at the “Plot” tab together with the options to zoom into it. For now, it is only possible to specify the ranges of values that should be visible in the plot; in future versions, however, the possibility to select a specific time period or a specific range in the currently loaded time labels will be present. The “restore all” buttons reset the ranges of x and y axes to their original values.

In the plot, each time series is drawn in a different colour. The colours are then assigned to the names of time series in a box in the top-right corner, which contains the legend.

When this plot is created, a set of basic statistics for each of the selected time series is output into the box in the bottom-left corner of the “Plot settings” tab, including the mean, standard deviation and median of each of the time series.

## 5.2. Interval time series plot

In the “Plot Settings” tab, it is possible to define one or several interval time series, specifying the time series that form their lower and upper bounds, or their centres and radii. The “Specify ITS” button brings up a dialog where there parameters can be defined using any of the loaded time series. The box to the right of this button then shows all the interval time series that the user has defined, in the notation [lowerBound, upperBound] or <centre, radius>, depending on how the ITS was created. Clicking on one or more ITS will select them (hold Ctrl to select more than one entry or Shift to select a range). The ITS can also be removed using the “Remove selected ITS” button.

The “Plot all specified ITS” button works in the same way as the “Plot selected time series” that was explained in the previous section. It produces a plot containing all the selected ITS, again scaled to the same scale. The plot can also be zoomed.

## 5.3. Scatterplots of the ITS Centre and Radius

Once an ITS is specified, a scatterplot can be produced for its centre and radius to see the degree of correlation between the two. The “Scatterplots all in one” button produces a scatterplot of centres vs. radii of all the selected ITS. Regardless if the ITS was specified as lower and upper bound or as centre and radius, the scatterplots always consider the values for centres and radii (they are calculated from the lower and upper bounds, if need be).

The “Scatterplot matrix” button produces a matrix of scatterplots for all combinations of centres and radii of the selected ITSs. On the diagonal of the matrix, a histogram of the particular quantity can be found, and above the diagonal, the R2 coefficient is shown.

This plot cannot be zoomed.

## 5.4. ACF and PACF plots

When one or more classical time series are selected in the box in the top-left corner of the “Plot settings” tab, the “Autocorrelation Plot” and “Partial Autocorrelation plot” buttons produce plots of ACF and PACF, respectively. These plots can be found in the box in the bottom-right corner of the “Plot settings” tab, organized into tabs, where each tab contains at most four plots.

These plots cannot be zoomed.

## 5.5. Boxplots, Histograms, Normal probability plots

Exactly the same as in the previous section, the “Boxplots”, “Histograms” and “Normal probability plots” buttons produce a particular kind of plot for all selected classical time series. The resulting plots will be placed in the box in the bottom-right corner of the “Plot settings” tab, organized into tabs, at most four plots on each of them.

## 5.6. Fitted and forecast values

For each forecasting method or model that is applied to the data, a plot of its fit to the training data and forecast values is produced. This is a special type of plot that will be discussed further in Section <TODO ref!> dedicated to running the forecasting methods.

## 5.7. Exporting the plots

The classical and interval time series plots can be exported using the “Save currently shown plot” in the “Plot” tab. A dialog will appear enabling the user to type in the requested name for the file and letting the user choose the format of the image. Currently, the following formats are supported: .eps, .ps, .png, .pdf. In case a file with the same name already exists in the destination folder, a warning is issued, asking the user about their intentions to overwrite the file.

# 6. Data transformation

In the “Data transformation” tab, several options are provided to transform the currently loaded data.

## 6.1. Difference time series

The “Difference selected” button will produce a first difference of all the time series that are selected in the box in the top-left corner of this tab. The resulting time series will be added to the list, so it is possible to difference an already differenced time series. In the current version, the differenced time series are named following this pattern: “DIFF(<originalTS>)”, where <originalTS> refers to the name of the time series that was differenced.

The user can work with the differenced time series exactly in the same way as with the originally loaded data, i.e. it can be used for plotting, analyses, forecasting etc.

## 6.2. Logarithmic transformation

The “Log transform selected” button will produce time series whose values are equivalent to the logarithm of the original values of the selected series. Their names will be “LOG(<originalTS>)”, where again <originalTS> stands for the name of the time series that was used for computing the logarithms.

The resulting time series are added to the currently loaded data and can be worked with exactly in the same way as the original time series.

6.3. Remove trend

The “Remove trend” button takes all the selected time series and for each of them produces a new time series where the regression line has been subtracted from the original values. The names of the new time series follow a general pattern of “NOTREND(<originalTS>)”.

The resulting time series are added to the currently loaded data and can be worked with exactly in the same way as the original time series.

All the data transformation methods can be combined.

# 7. Basic analyses of the data

## 7.1. Basic statistics

As it was mentioned in Section <TODO ref>, the mean, standard deviation and median of each classical time series is output when plotting the series.

## 7.2. Normality tests

The “Tests for normality” button on the “Plot settings” tab conducts a basic analysis of all the selected classical time series and outputs the results into the box in the bottom-left corner. The normality tests used in this analysis are the following: Anderson-Darling, Cramer-von Mises, Lilliefors (Kolmogorov-Smirnov), Pearson chi-square, and Shapiro-Francia test. For each test, the p-value is output.

## 7.3. Stationarity tests

The “Tests for stationarity” button on the “Plot settings” tab conducts a basic analysis of all the selected classical time series and outputs the results into the box in the bottom-left corner. The stationarity tests used in this analysis are the following: Ljung-Box, augmented Dickey-Fuller, and Kwiatkowski-Phillips-Schmidt-Shin test. For each test, the p-value is output.

# 8. Forecasting methods and models

The tool currently supports the following forecasting methods and models: the multi-layer perceptron (MLP), the interval multi-layer perceptron (iMLP), radial basic functions neural network, ARIMA, k-nearest neighbours, vector autoregressive model (VAR), simple exponential smoothing (SES), Holt’s method, Holt-Winters’ seasonal method and a hybrid approach that allows to combine any of the aforementioned.

The settings for all these methods can be found in the “Analysis settings” tab. For each method, a movable slider at the top of the settings indicates the portion of data that will be used for the training phase. The rest of the data will be used for testing the accuracy of the method/model. The sliders are independent among the different methods/models, but shared among different implementations of the same model, if there is more than one available. The current value selected on the slider is displayed in the box to the right of it. The box and the slider are synchronized, i.e. if a percentage is input into the box and confirmed by pressing enter, the slider will be reset according to that value.

The following sections describe the individual parts of the “Analysis settings” tab pertaining to the particular methods.

## 8.1. MLP

Two R packages are implemented for the MLP: nnetar and nnet. Each of them has a slightly different set of parameters that can be set and their performance differs as well.

### 8.1.1. nnetar

As its name suggests, the package is intended for autoregression, i.e. a variable under examination is explained by its own past values. It allows to specify the number of nodes in the hidden layer (only one hidden layer is supported), the number of seasonal and non-seasonal lags to be used as the input, the Box-Cox lambda parameter and the number of networks that will be trained. Out of the trained networks, the best one is chosen and shown to the user. All these parameters have their default values, so the user is not obliged to input anything in the basic scenario.

### 8.1.2. nnet

The package nnet provides an extended set of settings, including explanatory variables. The values for a particular explanatory variable can be selected from all the time series that are currently available. It is also possible to specify a lag that will be used, and optionally a custom name for the variable in case the original name of the time series is confusing. If a name for the variable is not specified, a placeholder name “Variable<number>” is used, where <number> is unique for each variable. All the specified explanatory variables are shown in the box and it is possible to select and remove one or several of them.

It is furthermore possible to specify the number of nodes in the hidden layer (in the current version, only one hidden layer is available), to add skip-layer connections from input to output, specify the range for the random initialization of the weights, the weight decay parameter, the maximum number of iterations used for the training of the network, the option to turn on trace optimization, and two stopping criteria – absolute and relative.

The R package provides even more options, but they are currently not implemented in the tool; mostly due to the difficulties in specifying the values of the parameters. In future versions, all the options will be available to the user.

## 8.2. MLP(i)

The MLP(i) approach, also known as the pseudo-interval approach, enables the user to obtain interval forecasts using a classical method. The user can specify the settings for two separate MLP networks (both using the same R package), one of which will be used for the centre and the other for the radius. The settings are the same as in the classical time series version mentioned in the previous section.

In addition to the general settings, it is possible to choose to train a number of networks. From these networks, only the best one is selected and shown to the user, and this is done based on the specified criterion. The possible criteria for selecting the best network are as follows: the highest coverage rate, the highest efficiency rate, the highest sum of the coverage and efficiency rate, the lowest MDE, the lowest RMSE, the lowest MSE, the lowest Theil’s U, the lowest ARV, the lowest RMSSE for the centre, and the lowest RMSSE for the radius. All the error measures mentioned among these criteria will be further discussed in Section <TODO ref>.

Finally, a distance measure can be selected and used for the computation of the accuracy of fit of the resulting network. Euclidean, Hausdorff, Ichino-Yaguchi, De Carvalho and Bertoluzza distances are available, and for each of them, values of the parameters (if any) can be input manually.

## 8.3. iMLP

For the iMLP (interval MLP), it is possible to add and remove explanatory variables in the same fashion as it was mentioned in Section <TODO ref>, with the slight difference in the sense that the variables are interval-values. For this reason, the specification of the variable requires to select the lower and upper bound, or alternatively, the centre and the radius. Furthermore, the output variables need to be specified, again as interval values. In the current version, only one output variable is supported, although the underlying implementation handles any number of output variables natively.

Other settings for the iMLP include: the number of neurons in the single hidden layer, the number of training iterations, the number of networks to train and the distance to use for training. Again, from the trained networks, only the best is shown based on the selected criterion (the criteria are the same as discussed in the previous section). The distance to use for the training is one of the following: Euclidean, Hausdorff, Ichino-Yaguchi, De Carvalho and Bertoluzza distance. The same distance will also be used for the calculation of the error measures. If several distances are selected, the corresponding number of instances of iMLP is created and solved separately. All the results can then be viewed and compared together.

## 8.4. RBF

Only the explanatory variables, the number of neurons in the hidden layer and the maximum number of training iterations need to be specified for this version.

## 8.5. RBF(i)

Analogously to the MLP(i), the RBF(i) consists of two separately configurable RBF networks, one for the centre and the other for the radius, and produces interval forecasts. It is possible to select the distance to use for computing the error measures as well as the number of networks to train together with the selection criterion.

## 8.6. ARIMA

The user can either input the values for the parameters (p, d, q, P, D, Q) manually, or select the “optimize parameters” checkbox and let the method find the optimal values for the particular data set. It is also possible to use the ARIMA model with constant by selecting “include constant”.

After selecting the “compute prediction intervals for the forecasts” checkbox, the specified prediction intervals will be computed and plotted for the forecasts based on this ARIMA model.

## 8.7. kNN

Two R packages are available for the computation of the k-nearest neighbours: kknn and FNN.

### 8.7.1. kknn

This package requires the lag of the data to be specified, and also the maximum number of neighbours to consider. It then finds the optimal number of neighbours up to the specified maximum.

### 8.7.2. FNN

This package requires the lag of the data to be specified and the number of neighbours to consider.

## 8.8. VAR(i)

The VAR model will be applied to the centre and radius of a chosen interval time series, producing interval forecasts. The input settings include the lag to consider, and possibly choosing to optimize the lag according to a selected criterion: AIC, HQ, SC, or FPE.

Please note: This model is currently in the experimental version and may not work as expected.

## 8.9. SES

The simple exponential smoothing model only allows to input the value for the alpha parameter. If no value is input or if the “optimize” checkbox is selected, the method will try to find an optimal value for this parameter.

## 8.10. Holt

The values for the alpha and beta parameters can be input manually. If no value is input or if the “optimize” checkbox is selected for any of the two parameters, the method will try to find an optimal value for this parameter. It is also possible to use the damped version of the Holt’s method by selecting the “damped” checkbox.

After selecting the “compute prediction intervals for the forecasts” checkbox, the specified prediction intervals will be computed and plotted for the forecasts based on this Holt model.

## 8.11. Holt-Winters

The values for the alpha, beta and gamma parameters can be input manually. If no value is input or if the “optimize” checkbox is selected for any of the three parameters, the method will try to find an optimal value for this parameter. It is also possible to use the damped version of the Holt-Winters’ method by selecting the “damped” checkbox, as well as choose the additive or multiplicative variant from the combo-box next to the gamma parameter.

This method requires the data to be seasonal, and it is necessary to specify the type of seasonality. The combo-box at the end of the settings allows the user to select among daily, monthly, quarterly and yearly seasonality.

## 8.12. SES(i), Holt(i), Holt-Winters(i)

These are the pseudo-interval variants of their respective methods. They work the same way as it was already explained for MLP(i) and RBF(i) – a separate model is configured and constructed for the centre and radius. A selected distance is used for the calculation of the error measures.

## 8.13. Hybrid

This is a generalized version of the pseudo-interval approach, allowing the user to forecast the centre and radius with completely different methods or models, configured independently on one another. The settings are always the same as in the individual methods.

## 8.14. A note on running multiple instances of the same method/model

Each numerical parameter P can take values in the following forms (specified by a formal grammar):

P -> N | P,P | P…P

where N is a single numerical value. Comma denotes enumeration, “…” denotes a range of values. For example, an expression in the form: “1,3,10…12,15” selects numbers 1, 3, 10, 11, 12, 15. If at least one of the bounds of the range is not integer, the range is treated as a range of non-integers, and a step of 0.01 is considered. For example, “0…0.04” takes numbers 0, 0.01, 0.02, 0.03, 0.04.

In cases like these, when multiple values are selected in any of the fields, a number of models corresponding to all combination of the specified settings is created. All of these models are then run separately and their outputs are shown to the user. The possibility to specify multiple values for a parameter is especially useful in cases where a sensitivity analysis is beneficial.

For some fields, however, it is not possible to specify multiple values. For this reason, the tool implements the functionality of batches. Each part of settings corresponding to a particular method or model has its own “Add to Analysis batch” button, which saves the currently selected settings so that user can set up a completely different set of parameters to run at the same time. Any number of settings can be saved like this, and then run at the same time using the “Run batch” button on the “Analysis batch” tab.

# 9. Run menu

After specifying the properties of each of the models the user wishes to use, the “Run” tab enables them to actually run the analyses. It is possible to observe the accuracy of a selected model/method, or additionally, to produce a number of forecasts specified in the field at the top. Please note that due to various reasons, one of them being the presence of explanatory variables, some methods and models are not able to produce forecasts as of yet.

The analyses can be run with the whole data set, or just with a part of it, as per the values stated in the fields at the top (“Run with data at positions X to Y”). Currently, it is not possible to denote the range in any other way, but in the future the user should be able to select a range of timestamps; for example, the data from the January 1, 2010, to March 25, 2011.

All the methods listed in the previous section are available here as checkboxes and additional two benchmarking models are included: the random walk for classical time series and its variant for interval time series. The methods and models are divided into groups based on their application – those that are used for classical time series are at the top, then those for the pseudo-interval approach, and finally the purely interval methods.

For the classical and pseudo-interval methods, the user can select the main data set in the combo boxes to the left. Methods and models that do not use explanatory variables or otherwise do not specify the data in the settings, will use these time series.

After the purely interval methods there are methods for the combination of forecasts. These include the simple average, the weighted average with weights inversely proportional to MDE or the Theil’s U statistic, median, and in the case of interval time series also the weighted average with weights directly proportional to the sum of coverage and efficiency, and the average of centre and logarithm of radius. Each of these approaches includes a variant combining only the instances of the same method/model, and a variant combining all the instances that are included in this run. The user can surely understand that the same size of the data sets that are combined by these approaches is essential, so any of these methods will output an error message in case the size of all inputs is not the same. In such case, the results for this combination method will not be computed.

The “do not show all plots, just the average” checkbox enables the user to de-clutter the resulting plot in case a large number of models/methods was run. Only the averages will be drawn, thus eliminating unnecessary lines in the plot.

The user can select any number of methods to run and then hit the “Run” button. All the analyses will be conducted and a number of different outputs will be produced:

For each group of methods/models (i.e. classical and interval – including pseudo-interval – methods), a plot is produced, including the original time series as well as the fitted and forecasted values for each method. (So a maximum of two plots will be produced.) Each of this plot can be zoomed in the same way as it way explained in Section <TODO ref>, and in addition, the checkboxes in the legend enable the user to turn off (and back on) specific lines in order to make the rest of the plot more clear. The training and testing set are separated by a vertical dashed line in the colour of the model/method, and the same line separates the testing set from the future forecasts for which there are no longer evidence data entries. If a particular method allows to compute prediction intervals and such option was selected, the prediction intervals for the forecasts for this method are highlighted in the same colour as the rest of the line. The bounds of the prediction intervals can be found in the “Prediction intervals” tab. This tab is only generated if some prediction intervals were requested.

Furthermore, diagrams of the structure of each neural network, if any, will be output to the “Diagrams of NNs” tab, and the residuals, forecast values and error measures will be printed on their respective tabs.

The “Residuals” tab lists the residuals for all the methods/models that were used in this run and allows to view their plots and basic statistics. Clicking inside the columns, it is possible to select one or more residual time series, and using the “Plot” button, the plot of all the selected ones together with their mean, standard deviation and median, will appear in the two boxes to the right.

The “Forecast values” tab lists the forecasts from all the methods/models, or states <not available> in case the particular method/model does not support forecasting yet. For the sake of clarity, the user can choose to hide a particular column either by double-clicking inside of it, or by using the filtering buttons at the top of this tab. The hidden columns can be shown again by clicking on the “Show hidden columns” button.

The error measures tab includes two tables for each group of methods that were run (classical or interval) – a table of the accuracy of the methods on the training data set, and the accuracy on the testing data set. For classical time series methods, the following error measures are considered: ME, RMSE, MAE, MSE, RMSSE, MPE, MAPE, MASE and Theil’s U. For interval time series methods, the following error measures are considered: MDE, RMSE, MSE, RMSSE (centre), RMSSE (radius), mean coverage, mean efficiency, Theil’s U (interval version), ARV (interval version). Note that in general, the values for MDE, RMSE, MSE and RMSSE for the interval methods cannot be compared easily, since they may have been produced using different distance measures. In these tables, a double-click inside a row again hides the selected row (and for consistency, also hides its counterpart in the other half of the table; e.g., hiding the results from the random walk in the training part will also hide the random walk in the testing part of the table). This option is especially useful when many models/methods are included in the table, for example after performing a sensitivity analysis, and the analyst can see that some of them are clearly useless (e.g., based on their high values for the Theil’s U). Hiding these rows clears the table and gives a better idea of the more useful cases.

The table of residuals, forecast values and the table of error measures can all be exported to an Excel spreadsheet. Note that if a column or row is hidden, it will not be exported; the export function only export the values that are visible to the user at that particular moment.

# 10. Analysis batch

(I am not sure if this needs further comments…)

# 11. Table of error measures

# 12. residuals

# 13. Forecast values

(These three (11, 12, 13) have already been described in section 9, but I can separate the explanations into these sections later.)

# 14. Diagrams of NN

(TODO: a few more words on this maybe?)