

## **METHOD AND SYSTEM FOR SCENARIO-BASED FORECASTING**

### **TECHNICAL FIELD**

The present disclosure relates to methods and systems for scenario-based forecasting.

### **BACKGROUND**

Scenarios are a common tool used in forecasting, as are predictions based on data extrapolations. Most computer-based forecasting techniques tend to provide a large margin of error to account for unforeseen scenarios. Further, such forecasting techniques do not allow users to interactively observe the variations in predictions based on their expected scenarios.

It is desirable therefore to overcome or alleviate one or more of the above difficulties, or at least to provide a useful alternative.

### **SUMMARY**

Disclosed herein is a computer-implemented forecasting method comprising:

- receiving, at a baseline forecaster module, time series data relating to respective time series for a plurality of respective variables, the time series data being derived from a plurality of data sources;

- generating, by the baseline forecaster module, one or more forecasts for future time points for each of the plurality of variables;

- obtaining a scenario forecaster configured to estimate future values of a selected variable based on the one or more forecasts for the remaining ones of the plurality of variables;

- receiving a user input of a scenario, the user input relating to values of the remaining ones of the plurality of variables; and

- using the scenario forecaster, estimating a value for the selected variable based on the values of the remaining ones of the plurality of variables.

Also disclosed herein is a forecasting system comprising:

- a baseline forecaster module configured to:

- receive time series data relating to respective time series for a plurality of respective variables, the time series data being derived from a plurality of data sources; and

- generate one or more forecasts for future time points for each of the plurality of variables; and
- a scenario forecaster configured to:

obtain a predictive model that is configured to estimate future values of a selected variable based on the one or more forecasts for the remaining ones of the plurality of variables;

receive a user input of a scenario, the user input relating to values of the remaining ones of the plurality of variables; and

using the predictive model, estimate a value for the selected variable based on the values of the remaining ones of the plurality of variables.

The selected variable may be land value.

The baseline forecaster module may be configured to generate the one or more forecasts by a neural network. The neural network may comprise one or more bidirectional long-short term memory (LSTM) layers. The baseline forecaster module may be configured to input, to the neural network, a feature vector that comprises one entry for each data source, each entry corresponding to a single normalized variable. A feature vector that is input to the neural network may comprise one entry for each data source, each entry corresponding to a single normalized variable.

The scenario forecaster may comprise a two-stage stacked ensemble model. A first stage of the two-stage stacked ensemble model may be configured to generate a plurality of predictions by a plurality of regression models. The plurality of regression models may comprise a gradient boosting regressor and a nearest neighbors regressor.

## **BRIEF DESCRIPTION OF THE DRAWINGS**

Embodiments of the present invention will now be described, by way of non-limiting example, with reference to the drawings in which:

Figure 1 is an example architecture of the system of the invention;

Figure 2 shows an example neural network architecture for the baseline forecaster module; and

Figure 3 shows cloud processes performed by the system of Figure 1.

## **DETAILED DESCRIPTION**

Embodiments of the present disclosure relate to a machine learning application that dynamically learns and extrapolates from existing as well as user-determined sets of conditions, and variable inputs, to

produce accurate long-term forecasts that account for ways in which the impact of input variables on the dependent variable may change over time.

In some embodiments, the method allows for feedback loops that take into account granular changes in macro-level factors including non-stationary variables in areas such as climate, infrastructure, demographics, energy, fiscal condition, socio-economic patterns, and many others.

In some embodiments the method may be used to forecast the value of land and property prices for example to assess the potential for appreciation and depreciation over time. The unit of land may be measured at the level of a subdivision of a geographical region, such as a state or a county. The method can be applied to different real estate asset classes not limited to residential, commercial, industrial or agricultural assets.

An architecture for implementing such methods is shown in Figure 1. The architecture describes a pipeline or flowchart of the present methods, from the ingestion of data to receiving user inputs and forecasting results. The architecture 100 forms a forecasting system 100 that in some embodiments comprises a data processing module 110 that is configured to receive time series data each relating to a respective time series for a plurality of respective variables. Two or more time series data may relate to a single time series, or each may relate to a different time series. The time series data are derived from a plurality of data sources 104. The time series data may relate to variables for a particular geographic region. For example, the variables may comprise variables in environmental, socio-economic, political, and other domains. At least one of the variables may be a real estate-related variable such as land value and/or property value. The variables may be of different types, for example numerical (integer or real-valued), categorical, or binary.

The data processing module 110 may comprise one or more components for ingesting the input data from data sources 104 and preparing it for input to a forecaster model 120 that is configured to generate one or more forecasts for future time points for each of the plurality of variables. In some embodiments the baseline forecaster model 120 is a machine learning model, and some examples of such machine learning models will be described in further detail below. The baseline forecaster model 120 may be a univariate model. That is, the forecasts for future time points are performed by forecaster model 120 for each variable separately over time, without accounting for correlations between the variables.

For example, the data processing module 110 may comprise a parser 112 that receives the different sources 104 of data that come in various formats. For example, statistical data may be received from data sources 104 in CSV or XLS-X format, while geospatial data may be received in Geotiff or Shapefiles

or Geojson format. The parser 112 loads the different types of/differently formatted data and translates them into structured data for input to the forecaster model 120. The structured data may take any suitable form, such as a tabulated dataframe for which each entry has a country code, date and other fields necessary for analysis.

The processed structured data contains multiple time series data grouped according to the number of targeted geographical regions. For example, if there are 3,000 counties, there will be 3,000 groups of data (e.g. each time series data may form a respective group) with the same interval and time period for a single variable.

The forecaster module 120 may also comprise a data cleaning component 114. The cleaning component 114 receives parsed structured data from the parser 112, and performs one or more preprocessing operations on the parsed data to improve the accuracy of the forecasts made by forecaster model 120. For example, transformer 114 may check the parsed data for null values and remove or flag these so that they can be handled appropriately by the forecaster model 120. The cleaner 114 may also pad null values using an average or median value across all values for each data group for the relevant time interval, or by interpolating within the same data group which may further include accounting for non-linear seasonal data variations. The cleaner component 114 may also detect and remove outliers from the parsed data. Further, the cleaner component 114 may normalize the parsed data, for example by standardization or min-max normalization.

The data processing module 110 may also comprise an exporter component 116 that splits the variables in the normalized parsed data into univariate data (i.e., one variable per data set) before passing the univariate data to the forecaster model 120.

The forecaster model 120 receives the normalized parsed data or data exported by the exporter component 116, and generates one or more predictions for future time points, based on historical time series data, separately for each variable. This results in forecast data 122 comprise the predicted future value or values (one or more predictions at one or more corresponding future time points) for each of the plurality of variables.

To improve the accuracy of the forecasted results, the forecaster model may employ upsampling layers and neural network layers that regularise the forecasted future trend.

Turning to Figure 2, an example neural network for implementing forecaster model 120 will be described. The neural network 200 first comprises an embedding layer 202 that groups the time-series

into sub-series based on a target. For example, the time series of different geographical regions may be grouped into sub-series that are specific to one geographical region only. This region might be at the level of State, County, CBSA, Zip Codes, or other geographical areas of concern.

The embedding layer 202 is connected to a data augmentation layer 204 that upsamples the number of datapoints of the embedded inputs while maintaining the historical trend. This layer helps to regularise the prediction and improve downstream accuracy. Multiple upsampling or data augmentation layers may be used where desired.

The upsampling layer 204 is then connected to a convolutional layer 206 (e.g. a 1-dimensional convolutional layer) with the appropriate kernel size. This layer 206 is in turn connected to 2 or 3 hidden fully connected layers 208, 210 after applying a pooling process at layer 212.

Afterwards, an output layer 214 forwards the data to a downsampling layer 216 that reduces the data points generated by the upsampling layer 204 into the original structure of the time series. In some embodiments, the output layer 214 can be omitted. The final output layer or output node 218 returns the forecasted result or results of the specified variable in the next year. The entire neural network 200 is built in an autoregressive loop 220 where the predictions are fed into the network 200, e.g. at the data augmentation or upsampling layer 204, until the desired length of forecasted results are reached.

Referring again to Figure 1, the forecasted results may then be transmitted from the baseline forecaster 120 to memory – e.g. forecasted baseline data 122 which may be a data store such as non-transitory memory. Many or all of the processes up to this point can be performed offline.

The forecasting system 100 also comprises a series of cloud processes that dynamically respond to the input of the user 102 and return forecasting results based on user input in real-time. In some embodiments, there may be a subset of the cloud processes of those specified here, or additional cloud processes not specified here. The cloud processes implement a scenario parser 140 that receives user input, determines weights to be applied to user selected features based on the user input, and outputs a modified input data to a scenario forecaster module 150. The cloud processes also implement the scenario forecaster module 150 which receives the modified input data from the scenario parser 140. The scenario forecaster 152 outputs a forecast adjusted based on the user input. This allows the user to visualize the effect of changes in input variables on the forecast or prediction – e.g. an increase in school funding for a particular area and the effect that increase has on property prices or demographics (e.g. young families moving to the area). The output of the scenario forecaster 152 can be visualized on the interface 130 as real time feedback to the user.

Turning to Figure 3, an example of the interface and cloud process will be described.

The user interface 131 enables the user 102 to increase or decrease the likelihood of certain predefined scenarios, or to adjust the values of individual independent variables to be used for generation of the scenario-based prediction 142. For example, the user 102 may adjust the time horizon of the prediction, a degree of climate change, an aging rate of the population in the geographic region (e.g. at state level or county level), a rate of population movement, a crime rate, and/or an amount of infrastructure spending. Further or different variables may be adjustable via the user interface 131 depending on the use case or availability of data for the feature or features.

The scenario parser 132 receives inputs from both the user interface 131 and forecasted baseline data 122. Some or all of the forecasted baseline data 122 may be displayed to the user 102 via user interface 131 as a baseline scenario. The baseline scenario is a projection of the future values of the plurality of variables without influence of the user, and can be adjusted by the scenario parser 140 in accordance with the input from the user interface 131. Accordingly, the scenario parser 140 outputs modified values 134 of the plurality of variables.

The user 102 may select one or more of the plurality of variables, such as land value, as a dependent variable. The remaining ones of the plurality of variables are then treated by scenario forecaster module 150 as independent variables.

The scenario parser 140 processes the user input by reading the user input and encoding the user input into different weights. The weights adjust the input dataset for prediction. The scenario parser 140 receives the forecast baseline data 122 and separates it based on the features learned by the machine learning model from features outside of the model. The scenario parser 140 distinguishes between features of the feature space learned by the machine learning model 120 and those adjustable by the user. It then separates the features into two sets and feeds the corresponding sets to either the machine learning pipeline 151 and the statistical model pipeline 150 of Figure 3. The sets of features are mutually exclusive. In other embodiments, the feature sets have overlapping features.

Features that are learned by the machine learning model have different degrees of correlation or feature importance. Adjusting the value of the variables thus affects the machine learning results. Features that are not learned by the model are modelled with statistical models that are calculated based on research and heuristics. For example, some climate hazard data such as "number of heat waves per year" might have a weak correlation with a target variable or result of a scenario. However, as climate change worsens, the effect of heat waves may more strongly correlate. This correlation increase needs to be assessed using statistical models.

The machine learning model 151 comprises an ensemble machine learning model that is configured to estimate future values of the selected dependent variable based on forecast data for the independent variables. For example, the machine learning model 151 may be obtained by training using the historical values of the independent variables and the forecast values of the dependent variables from the forecast baseline data 122. In some embodiments, the machine learning model 151 is obtained by retrieving stored parameters of the model from a database of previously trained models.

The machine learning model 151 is an ensemble machine learning model that relates the independent variables to the dependent variable. The ensemble machine learning model employs decision tree components to consider the relationships of the dependent variable through class labels and a regression model the regressors of which are used to project values beyond the domain of the training dataset. The machine learning model may be trained by dividing the input data into a 70/15/15 Training/Validation/Test Split, with 5-fold cross validation.

In some embodiments, a series of preprocessing steps may be performed by the machine learning model 151. For example, the input data may be log-transformed and standardized. A null value for a variable in the input data may be replaced with the mean value for the variable. Categorical data may be transformed by one-hot encoding, or if appropriate, ordinal encoding.

During inference the modified values of the independent variables are passed to the scenario forecaster 150 to obtain a scenario-based prediction 152 of the dependent variable (e.g., land value in the geographic region of interest). The final result may be obtained by applying statistical weights to the machine learning results. For example, where a feature has been varied by a user it will not be part of the feature space of the machine learning model. In this instance, a statistical model (e.g. heuristic model) will be employed for scenario based forecasting by modifying the user-variable feature(s). The statistical model may be applied based on a relationship between a user-variable feature and the machine learning feature space, such as calculating a correlation between the non-machine learning feature (user-variable feature) and target variable using a Z-score – e.g. of a target county (user-variable feature) against the national database (machine learning feature space). The higher the deviation from the national mean, the more impact the feature will have on the final target price.

Throughout this specification and the claims which follow, unless the context requires otherwise, the word “comprise”, and variations such as “comprises” and “comprising”, will be understood to imply the inclusion of a stated integer or step or group of integers or steps but not the exclusion of any other integer or step or group of integers or steps.

The reference in this specification to any prior publication (or information derived from it), or to any matter which is known, is not, and should not be taken as an acknowledgment or admission or any form of suggestion that that prior publication (or information derived from it) or known matter forms part of the common general knowledge in the field of endeavor to which this specification relates.

**CLAIMS**

1. A computer-implemented forecasting method comprising:
  - receiving, at a baseline forecaster module, time series data relating to respective time series for a plurality of respective variables, the time series data being derived from a plurality of data sources;
  - generating, by the baseline forecaster module, one or more forecasts for future time points for each of the plurality of variables;
  - obtaining a scenario forecaster configured to estimate future values of a selected variable based on the one or more forecasts for the remaining ones of the plurality of variables;
  - receiving a user input of a scenario, the user input relating to values of the remaining ones of the plurality of variables; and
  - using the scenario forecaster, estimating a value for the selected variable based on the values of the remaining ones of the plurality of variables.
2. A computer-implemented forecasting method according to claim 1, wherein the selected variable is land value.
3. A computer-implemented forecasting method according to claim 1 or 2, wherein the baseline forecaster module is configured to generate the one or more forecasts by a neural network.
4. A computer-implemented forecasting method according to claim 3, wherein the neural network comprises one or more bidirectional long-short term memory (LSTM) layers.
5. A computer-implemented forecasting method according to claim 3 or 4, wherein a feature vector that is input to the neural network comprises one entry for each data source, each entry corresponding to a single normalized variable.
6. A computer-implemented forecasting method according to any one of claims 1 to 5, wherein the scenario forecaster comprises a two-stage stacked ensemble model.
7. A computer-implemented forecasting method according to claim 6, wherein a first stage of the two-stage stacked ensemble model is configured to generate a plurality of predictions by a plurality of regression models.
8. A computer-implemented forecasting method according to claim 7, wherein the plurality of regression models comprises a gradient boosting regressor and a nearest neighbors regressor.
9. A forecasting system comprising:

a baseline forecaster module configured to:

receive time series data relating to respective time series for a plurality of respective variables, the time series data being derived from a plurality of data sources; and

generate one or more forecasts for future time points for each of the plurality of variables;

and

a scenario forecaster configured to:

obtain a predictive model that is configured to estimate future values of a selected variable based on the one or more forecasts for the remaining ones of the plurality of variables;

receive a user input of a scenario, the user input relating to values of the remaining ones of the plurality of variables; and

using the predictive model, estimate a value for the selected variable based on the values of the remaining ones of the plurality of variables.

10. A forecasting system according to claim 9, wherein the selected variable is land value.
11. A forecasting system according to claim 9 or 10, wherein the baseline forecaster module is configured to generate the one or more forecasts by a neural network.
12. A forecasting system according to claim 11, wherein the neural network comprises one or more bidirectional long-short term memory (LSTM) layers.
13. A forecasting system according to claim 11 or 12, wherein the baseline forecaster module is configured to input, to the neural network, a feature vector that comprises one entry for each data source, each entry corresponding to a single normalized variable.
14. A forecasting system according to any one of claims 9 to 13, wherein the scenario forecaster is a two-stage stacked ensemble model.
15. A forecasting system according to claim 14, wherein a first stage of the two-stage stacked ensemble model is configured to generate a plurality of predictions by a plurality of regression models.
16. A forecasting system according to claim 15, wherein the plurality of regression models comprises a gradient boosting regressor and a nearest neighbors regressor.