DOI: 10.17323/2587-814X.2020.2.7.20

Transfer learning and domain adaptation based on modeling of socio-economic systems

Oleg D. Kazakov
E-mail: kod8383@mail.ru

Olga V. Mikheenko
E-mail: miheenkoov@mail.ru

Bryansk State Technological University of Engineering
Address: 3, Stanke Dimitrov Avenue, Bryansk 241037, Russia

Abstract

This article deals with the application of transfer learning methods and domain adaptation in a recurrent neural network based on the long short-term memory architecture (LSTM) to improve the efficiency of management decisions and state economic policy. Review of existing approaches in this area allows us to draw a conclusion about the need to solve a number of practical issues of improving the quality of predictive analytics for preparing forecasts of the development of socio-economic systems. In particular, in the context of applying machine learning algorithms, one of the problems is the limited number of marked data. The authors have implemented training of the original recurrent neural network on synthetic data obtained as a result of simulation, followed by transfer training and domain adaptation. To achieve this goal, a simulation model was developed by combining notations of system dynamics with agent-based modeling in the AnyLogic system, which allows us to investigate the influence of a combination of factors on the key parameters of the efficiency of the socio-economic system. The original LSTM training was realized with the help of TensorFlow, an open source software library for machine learning. The suggested approach makes it possible to expand the possibilities of complex application of simulation methods for building a neural network in order to justify the parameters of the development of the socio-economic system and allows us to get information about its future state.

Key words: transfer learning; domain adaptation, simulation modeling; decision support systems; socio-economic development of regions.

Citation: Kazakov O.D., Mikheenko O.V. (2020) Transfer learning and domain adaptation based on modeling of socio-economic systems. *Business Informatics*, vol. 14, no 2, pp. 7–20.
DOI: 10.17323/2587-814X.2020.2.7.20

Introduction

Management of the development of social and economic systems is mainly based on documents containing the planned values of indicators on a particular topic (strategy, concept, forecast, etc.). To date, the management of the region is carried out through monitoring by adjusting the planned values in accordance with those actually achieved [1]. This means that the basis for future development for the most part lies in the indicators of past periods obtained with a significant delay, if we take into account the real situation with the publication of official statistics. In this regard, the development of tools to justify the values of forecast economic parameters, which allows us to achieve planned control figures with a high degree of reliability, is an important scientific task. This task ultimately acts as an objective condition for the implementation of an effective economic policy.

The basis of the whole set of methods of socio-economic forecasting is traditionally made up of statistical methods used to build appropriate models of time series [2, 3]. Among the most common methods for analyzing time series are the following [4]: regression forecasting models (multiple and non-linear regression), exponential smoothing (ES) models, maximum similarity sampling model (MMSP), the Markov chains model, the Markov chains model on classification and regression trees (CART), a model based on the genetic algorithm (GA), a model on support vectors (SVM). The widest and most applicable of the classes of models are autoregressive forecasting models (ARIMAX, GARCH, ARDLM).

Recently, deep machine learning methods, whose quality metrics are much better than classical methods, have proven their effectiveness. However, the use of such models requires a huge amount of tagged data, which in real conditions it is not always possible to obtain. At the same time, when forecasting most of the indicators characterizing socio-economic systems and processes, statistical data are used for one decade and, in the best case, by month. In other words, there are only a hundred marked entries at the input. The problem could be solved in one way or another if teaching without a teacher was applied, which, unfortunately, at this stage of development of serial computer systems cannot be implemented in practice.

To solve this problem, we proposed to use a recurrent neural network built on the architecture of long short-term memory (LSTM) and trained on synthetic data obtained as a result of simulation with subsequent transfer learning (transfer learning) and domain adaptation (domain adaptation). By having real statistics for several decades, this will allow us, with a high degree of accuracy, to predict the values of economic parameters taking into account modern development vectors. Decision support systems based on these algorithms make it possible to most accurately justify economic plans and forecasts for the development of territories and ensure the achievement of strategic development guidelines.

1. Methods

1.1. Transfer learning and domain adaptation at LSTM

The main idea of transfer training is to solve the problem on the basis of "ready data" obtained as a result of solving similar problems. This means that you can first train a neural network on a large amount of data, and subsequently retrain it on a specific target set. In this regard, there are two main advantages of using transfer training [5]:

✦ a significant reduction in time and costs in the context of using the appropriate infrastructure for training, by training only a certain part of the final model;

✦ increasing the efficiency of the final model through the use of models trained on available data.

The results of this study are closely related to the second of these advantages, since in the predictive analysis of socio-economic systems this is a determining factor.

As the available data, synthetic data obtained as a result of simulation were used. Simulation is an experimental way to study reality using a computer model [6]. In simulation models, real economic processes are described as if they were actually happening [7]. Thus, simulation models can be used to study real socio-economic systems under the condition that economic objects and processes are replaced by a set of mathematical dependencies that determine what state the system will go from initially set [8].

The weights from the model trained on synthetic data obtained as a result of simulation are transferred to a new model. For this, the authors used the TensorFlow open machine learning software library.

The methods of transfer training and domain adaptation, as a rule, depend on machine learning algorithms used to solve the tasks [9]. One of the most effective tools for predictive analytics of socio-economic systems is recurrent neural networks with long short-term memory (LSTM networks). In particular, models based on the LSTM architecture are very effective for forecasting the time series — one of the most common tasks in managing socio-economic systems [10]. It should be noted that this efficiency does not decrease when predicting several steps.

The basic architecture of the recurrence network, developed back in the 1980s, is built from nodes, each of which is connected to all other nodes. For training with a teacher with discrete time, data is supplied to the input nodes at each next step. In this case, other nodes (output and hidden) complete their activation and the output signals are prepared for transmission to neurons of the next level [11]. Thus, a recurrent network with long-term memory allows use of information received in the past to solve current problems. In particular, it makes it possible to predict the values of the time series, since it does not use the activation function inside its recurrent components, and the stored value does not blur in time (*Figure 1*) [12, 13].

The LSTM module has five main components that allow it to simulate both long-term and short-term data [13]:

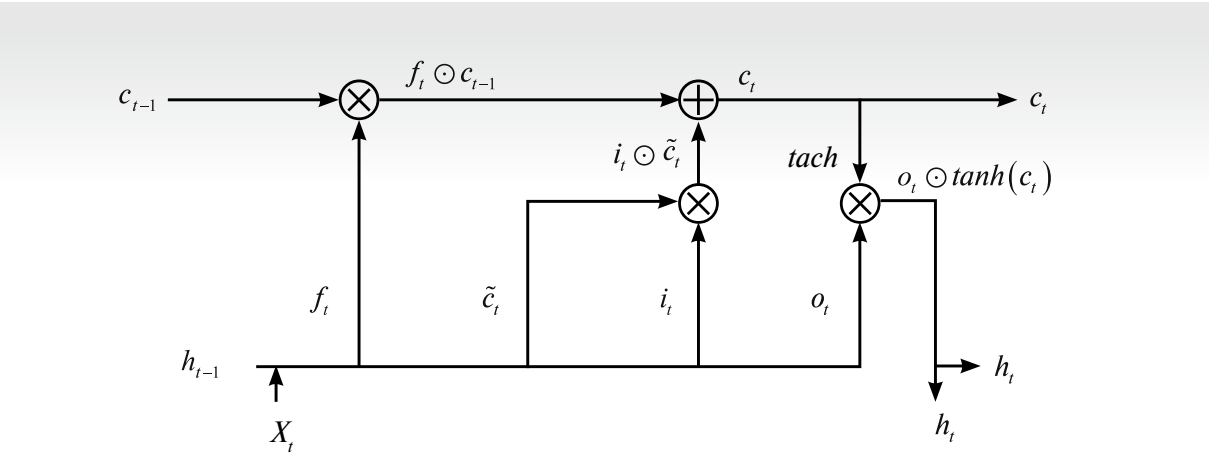


Fig. 1. Architecture of LSTM [12, 13]

$$\begin{cases} f_t = \sigma\left(W_f\left[h_{t-1}, x_t\right] + b_t\right) \\ i_t = \sigma\left(W_i\left[h_{t-1}, x_t\right] + b_i\right) \\ o_t = \sigma\left(W_o\left[h_{t-1}, x_t\right] + b_o\right) \\ \tilde{c}_t = tanh\left(W_c\left[h_{t-1}, x_t\right] + b_c\right) \\ c_t = f_t \odot c_{t-1} + i_t \odot c_t \\ h_t = o_t \odot tanh\left(c_t\right) \end{cases}, \quad (1)$$

where c_t – "state of the cell", representing its internal memory, which stores both short-term and long-term information;

h_t – "hidden state": such information about the output state, which is calculated by the current input, the previous hidden state and the current input of the cell, which will be used to predict one or another time series. The latent state may decide to extract short-term or long-term or both types of information from stored in (c_t);

i_t – "entrance gate": determine the amount of information coming from the current input in (c_t);

f_t – "transitional gate": determine the amount of information flowing from the current and previous (c_{t-1}) inputs to the current (c_t);

o_t – "exit gate": determine the amount of information falling from the current (c_t) into a hidden state.

Suppose there is a well-functioning model for predictive time series analytics "Kazakov_LSTM.h5" (the process of its training is presented in the next section of the article). Then, to view the parameters of this model, you can use the following instructions (*Figure 2*).

Thus, we obtain the following conclusion:

```
Parametr lstm_3_W_i:
[[ 0.00075, ...]]
Parametr lstm_3_U_i:
[[ 1.090001, ...]]
Parametr lstm_3_b_i:
[ 0., ...]
Parametr lstm_3_V_c:
[[-1.17770085, ...]]
```
(2)

where *W*-matrices – matrices that transform the input data;

U-matrices – matrices that transform the previous hidden state into another internal value;

b-vectors – offset for each block;

V – a vector that determines what values to derive from the new internal state.

The concept of domain adaptation is closely related to transfer training. The essence of this adaptation is to train the model on data from the source domain so that it shows comparable quality on the target domain [14]. The source domain can be synthetic data that can simply be generated by running the corresponding simulation model, and the target domain is a time series that reflects the dynamics of certain key indicators of the socio-economic system. Then the task of domain adaptation is to train the model on synthetic data, which will work well with real objects.

```
my_new_model = tf.keras.models.load_model('Kazakov_LSTM.h5')
```

```
for i in zip(my_new_model.layers[0].trainable_weights,
             my_new_model.layers[0].get_weights()):
  print('Parametr %s:\n%s' % (i[0], i[1]))
```

Fig. 2. Listing "LSTM parameter output"

The stage of domain adaptation is reduced to freezing weights in the "Kazakov_LSTM.h5" model in their previously prepared state. Domain adaptation weights are trained on the target data set. For this purpose, in the model after LSTM we add fully connected (dense) layers.

1.2. System-dynamic modeling of indicators of innovative development of socio-economic systems

In order to form a data set within the source domain, we will build a system-dynamic model that allows us to determine the parameters of the socio-economic system, in particular, to evaluate the values of the indicators of innovative development of the regions. The main document setting strategic guidelines for state policy in the field of innovative development in order to counter modern global challenges and threats is the Innovation Development Strategy of the Russian Federation for the period until 2020 [15]. The strategy therein determines the long-term development priorities of all subjects of innovation, and also sets a number of target indicators, which, in accordance with the installation of the Government of the country, should be taken into account when developing concepts and programs for the socio-economic development of Russia and its regions.

The strategy defines the values of target indicators for 2020. At the same time, 2010 is fixed as the base year, and 2013 and 2016 are intermediate control points. An analysis of the actual values of most of the target indicators for 2016 revealed a general tendency to lag behind the planned level.

Since statistical services prepare analytical data for the reporting period with a significant time lag, a serious problem is the fact that the state authorities responsible for the implementation of the Strategy are trying to develop managerial decisions, focusing on irrelevant performance results. It is extremely difficult to call such a process effective management.

In our opinion, the reverse movement will be the most effective approach when the value of a specific target indicator for a certain date is differentiated among the subjects of the federation and communicated to the regional authorities in advance, in the form of recommended forecast values. In this case, local government services will become direct participants in the process of implementing national strategic initiatives, including taking into account certain responsibilities for failure to achieve targets. In addition, it will be possible to manage on the basis of an up-to-date map reflecting the innovative development of the long-term objectives of the Strategy by regions.

The main components of the system model, which allows us to determine the innovative development of the region in accordance with the state strategy, are strategic tasks in key areas. The relationship between the final indicator of innovative development and these components (subindexes) can be described as follows:

$$I = \sum_{j=1}^{m} w_j \cdot I_j, \qquad (3)$$

where I_j – j-th subindex value;

m – number of subindexes;

w_j – weight factor of the j-th subindex.

Subindexes are summary indicators reflecting the formation of primary indicators as part of the solution of a specific strategic task in priority areas of innovation. The number of primary indicators in directions varies from 2 to 12, and each of them can be considered as an independent complex system-dynamic model. Consider the model for the formation of a first-order private indicator "Inventive Activity Coefficient," which is determined in the framework of the priority strategic task "Innovative Business." Models of other indicators can be formed in a similar way and are not presented in the framework of this article due to the significant scale of the study.

The system-dynamic model of the level of inventive activity is presented in *Figure 3*. The model is based on determining the ratio of the number of patent applications filed by domestic inventors to the total population. The number of patent applications developed and filed depends on the number of organizations engaged in research and development, the number of personnel involved in research and development, as well as the amount of internal expenses of organizations on research and development.

The corresponding mathematical model can be represented as follows:

$$\frac{d(\text{Inventions\_developed})}{dt} = \text{- patent}$$

$$\frac{d(\text{Useful\_Models \_Developed})}{dt} = \text{= - Useful\_Models\_Filed}$$

$$\frac{d(\text{patent})}{dt} = \text{Inventions\_submitted + Utility\_models\_submitted}$$

$$\frac{d(\text{population})}{dt} = \text{Birth - Death + Migration - Emigration} \quad (4)$$

$$\text{Inventions(Utility\_models) \_developed} = \text{corp\_research} \times \text{person\_research} \times \text{Internal\_R \& D costs}$$

$$\text{Death} = \text{Death\_trudospos\_zhazhra + Death\_ infantile + other}$$

$$\text{Migration} = \text{Refugees + Temporary shelter + Forced migrants + Arriving coeff\_inv\_activity} = \frac{\text{patent}}{\text{population}} \times 10\,000.$$

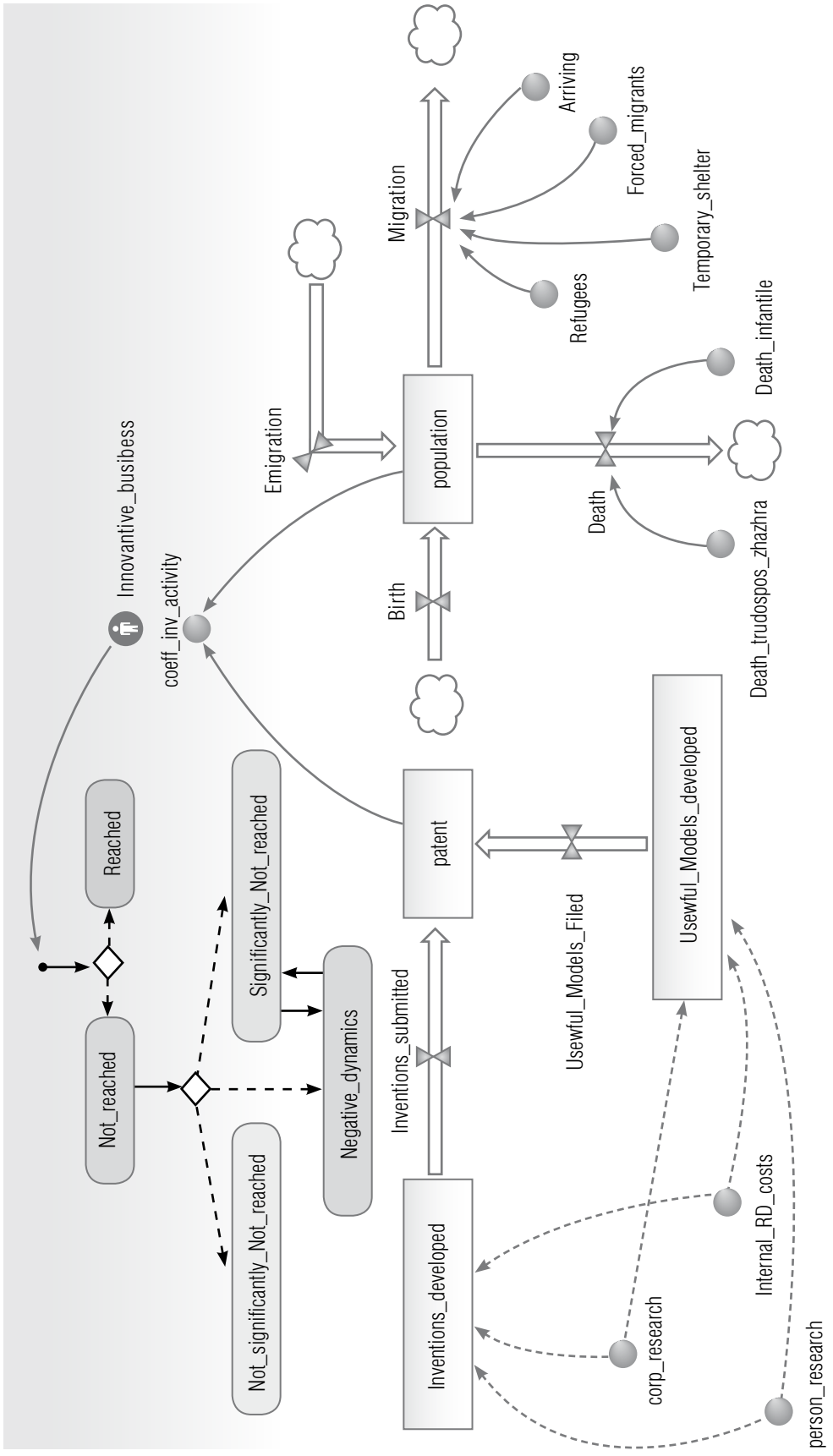
At the first consideration, the question arises of the appropriateness of using simulation to assess the level of inventive activity, since each of its components can be predicted (for example, within the framework of ARIMA or GARCH models that have proven themselves in the field of forecasting demographic and socio-economic indicators). In fact, the dependence of inventive activity on many indicators is stochastic; moreover, in practice, for most of them it is not possible to collect a sufficiently large set of values. Therefore, simulation in this context is considered as a way to build a model of a socio-economic system that describes the complex behavior of objects and processes associated with innovation management at the regional level. This model can be implemented any number of times. In this case, the results will be due to the random nature of the processes [16]. Using these results, one can obtain stable synthetic statistics on the level of inventive activity, which is subsequently used to train the neural network.

2. Experiment

2.1. Synthetic data generation using a system-dynamic model

Simulation modeling was implemented using a set of mathematical tools and AnyLogic special software, which allowed for targeted modeling in the "simulation" mode of the indicator under study, as well as optimization of some of its parameters [17]. In accordance with the results of a study conducted by a group of scientists from Kazan Technical University [18], the reliability of the AnyLogic system was found to be satisfactory, and in the ranking of similar software this system is among the top three.

The configuration settings of the model that graphically describes the user-posed problem in terms of the AnyLogic language are set using experiments. Discrete event modeling implements the possibility of approximating real processes by discrete events that consider the most important moments of the life of the simulated system [19].



coeff_inv_activity – coefficient of inventive activity, units;
patent – the number of domestic patent applications filed for inventions, units;
population – population of the territory, thousand people;
corp_research – number of organizations performing research and development, units;
person_research – the number of personnel engaged in research and development, people

Fig. 3. System–dynamic model of the level of inventive activity

An experiment "Variation of parameters" was carried out in the AnyLogic system, the essence of which was the repeated launch of the constructed simulation model. For the experiment, a confidence probability of 0.95 and an accuracy of 0.01 were determined. The number of model runs calculated by the Laplace function was 9604 [20]. Varying different parameter values, the model produced a label value ranging from 1.4725 to 2.1105. The mathematical expectation was 1.8114, and the dispersion of values relative to the mathematical expectation was 0.1539, which is acceptable and allows us to conclude that the proposed simulation model was successfully validated. The results of the experiment are presented in *Table 1*.

Table 2 presents the synthetic data obtained as a result of the experiment "Variation of parameters" in the AnyLogic system and used to train the primary neural network.

The initial data set contains five functions, the change in time of which is presented in *Figure 4*.

The figure shows that all time series have the property of seasonality, but we will not take this factor into account explicitly in further training of the original network.

2.2. Source LSTM training and learning transfer

As noted above, the training of the initial LSTM is carried out on synthetic data obtained as a result of simulation. To do this, use the TensorFlow open source machine learning software library, which provides a good helper application programming interface (RNN API) for implementing predictive time series models.

First of all, to complete the training process, we will load the synthetic data and standardize the data set using the mean () and std () functions.

Further, the task is reduced to predicting a multidimensional time series based on some provided history. We will create training and validation data and perform direct training of the original LSTM.

The multivariate_data function performs the window management task. It selects past observations based on a given step size (*Figure 5*). Next, we fix the weights of a pre-trained neural network (*Figure 6*). Then we create a composite neural network based on "Kazakov.h5" and compile it (*Figure 7*).

To select the best neural network hyperparameters, the Keras Tuner optimizer developed by the Google team and included in the Keras open library was used. RandomSearch was defined as

Table 1.

Results of the "Variation of Parameters" experiment of a system-dynamic model of the level of inventive activity in the AnyLogic system

Name	population	patent	corp_research	person_ research	coeff_inv_activity
Parameter Change	[135600…154235]	[21627…30732]	[3317…4384]	[672493…932115]	[1.4725…2.1105]
Expected value	144705.3752	26204.7906	3782.2631	778989.9941	1.8114
Dispersion	6347086.0729	4467795.1931	46155.5052	3873708843	0.0237
Standard deviation	2519.3424	2113.7160	214.8383	62239.1263	0.1539

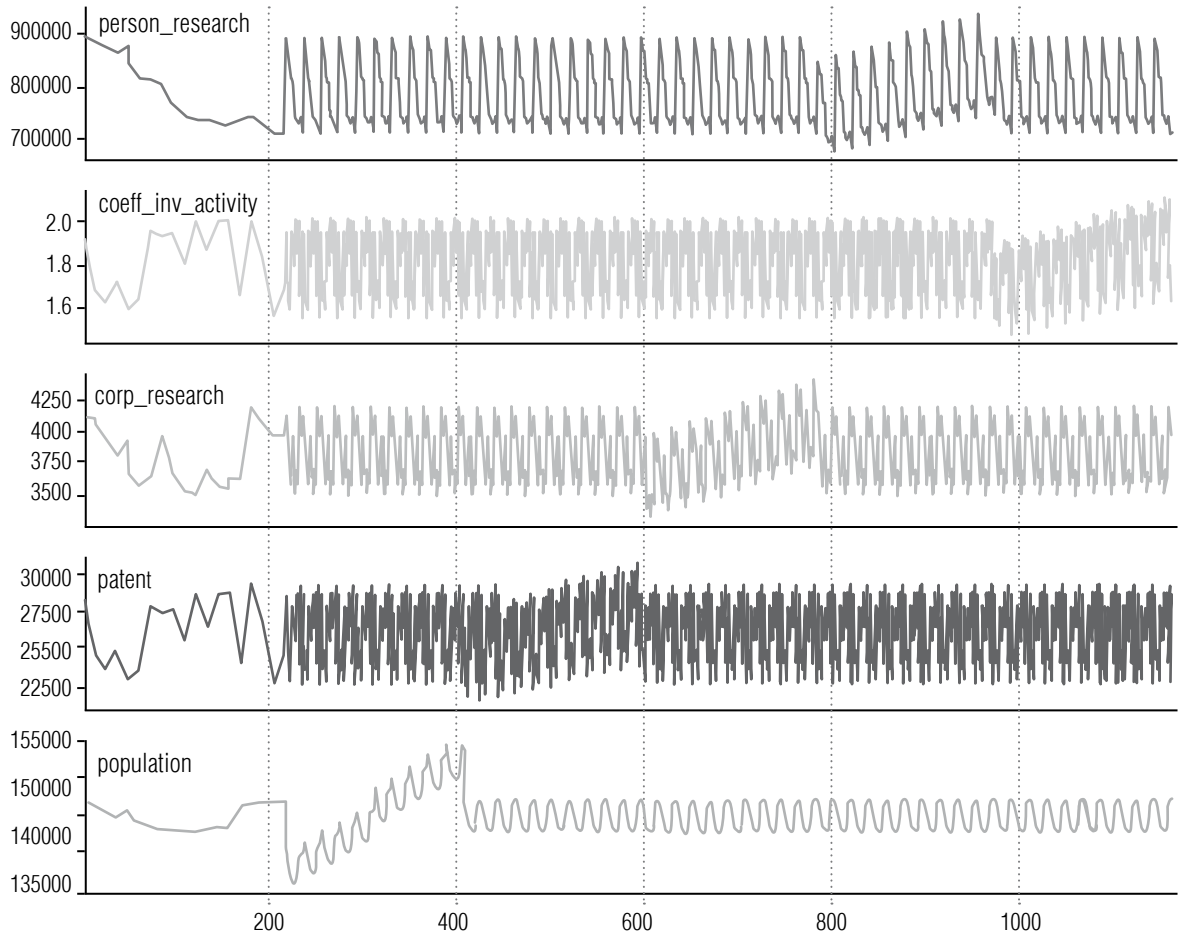


Fig. 4. The change in time of the original functions

```
x_train_single, y_train_single = multivariate_data(dataset, dataset[:, 1], 0,
                                                   TRAIN_SPLIT, past_history,
                                                   future_target, STEP,
                                                   single_step=True)
x_val_single, y_val_single = multivariate_data(dataset, dataset[:, 1],
                                               TRAIN_SPLIT, None, past_history,
                                               future_target, STEP,
                                               single_step=True)
```

```
single_step_model = tf.keras.models.Sequential()
single_step_model.add(tf.keras.layers.LSTM(32,
                                           return_sequences=True,
                                           input_shape=x_train_multi.shape[-2:]))
single_step_model.add(tf.keras.layers.LSTM(16, activation='relu'))
single_step_model.add(tf.keras.layers.Dense(1))

single_step_model.compile(optimizer=tf.keras.optimizers.RMSprop(clipvalue=1.0),
                          loss='mae')
```

Fig. 5. Listing “Training the original LSTM”

Table 2.

Synthetic data for training the original neural network

No	Signs				Mark
	population	patent	corp_research	person_ research	coeff_inv_activity
1	146890	28688	4099	887729	1.950000
2	146841	28362	4098	887553	1.931477
3	146792	28036	4097	887377	1.909913
4	146743	27710	4096	887201	1.888335
5	146694	27384	4095	887025	1.886743
…	…	…	…	…	…
9601	143267	24072	3604	732274	1.732500
9602	146545	29269	4175	738857	2.100000
9603	146804	26795	4032	722291	1.921500
9604	146880	22765	3944	707887	1.627500

the main type of Keras Tuner. Listing the best model with Keras Tuner is as follows (*Figure 8*).

To create a neural network with enumeration of the basic values of hyperparameters, the following function was used (*Figure 9*).

For two fully connected layers, Keras Tuner defined relu, the rectifier, and adam, the stochastic gradient descent method, based on an adaptive estimation of first and second order moments, as an activation function. The root mean square error (mse) is presented as a loss function, and the mean absolute error (mae) is used as a quality metric. In the last era of training, these parameters took the values of 0.3995 and 0.1739, respectively.

Thus, two fully connected layers were added for the implementation of domain adaptation based on actual data obtained from official statistics and presented in *Table 3*.

It is assumed that the prediction of the dynamics of inventive activity will be carried out by one step, so one neuron will remain at the output of the last layer of the network.

3. Discussion of the results

The developed simulation model of the dynamics of inventive activity allows you to create a potentially unlimited number of records for training the source network. The studied methods of transfer training and domain adaptation in LSTM allowed use of the pre-trained source network in the new mixed architecture. Thus, despite the available critically small set of evidence for training the neural network, it is possible to forecast economic indicators.

Using a trained neural network, we visualize the predicted values of the innovation activity coefficient for 2012 (based on data for the validation sample) and for 2018 (based on data for the test sample) (*Figure 10*).

```
my_new_model = tf.keras.models.load_model('Kazakov_LSTM.h5')
my_new_model.trainable = False
```

Fig. 6. Listing "Assigning weights from a pre–trained neural network"

```
single_step_model = tf.keras.models.Sequential()
single_step_model.add(my_new_model)
single_step_model.add(tf.keras.layers.Dense(14))
single_step_model.add(tf.keras.layers.Dense(1))
```

Fig. 7. Listing "Creating the architecture of a composite neural network"

```
[ ]  tuner = RandomSearch(build_model)
     tuner.search(x_train,
                  y_train,
                  epochs=20,
                  verbose = 1)
     models = tuner.get_best_models(num_models=1)
```

Fig. 8. Listing "Selection of neural network hyperparameters"

```
def build_model(hp):
    model = Sequential()
    activation_choice = hp.Choice('activation', values=['relu', 'sigmoid', 'tanh'])
    model.add(Dense(units=hp.Int('units_input',
                                 min_value=7,
                                 max_value=30),
                    activation=activation_choice))
    model.add(Dense(1, activation=activation_choice))
    model.compile(
        optimizer=hp.Choice('adam', values=['adam','rmsprop','SGD']),
        loss='mse',
        metrics=['mae'])
    return model
```

Fig. 9. Listing "Function for selecting neural network hyperparameters"

The data obtained for 2012 showed the value of the mark 1.91 with a value of 2.00 actually recorded during this period (Figure 10a). The inventive activity coefficient determined by the network for 2018 is 1.73 against the actual 1.70 (Figure 10b). Thus, the deviation was 4.5% in 2012 and 1.8% in 2018, respectively. The results can be considered satisfactory, which allows us to broadcast this method in the future.

Conclusion

The approach presented in the study, based on the construction of a system-dynamic model and a recurrent neural network, can be adapted

Table 3.

Evidence for implementing domain Adaptation

No	Year	Signs				Mark
		population	patent	corp_research	person_ research	coeff_inv_activity
0	2001	146304	24777.0	4037	885568	1.69
1	2002	145649	23712.0	3906	870878	1.63
2	2003	144964	24969.0	3797	858470	1.72
…	…	…	…	…	…	…
15	2016	146804	26795.0	4032	722291	1.83
16	2017	146880	22765.0	3944	707887	1.55
17	2018	146781	24952.8	3944	707887	1.70

Compiled on the basis of the official website of the Federal State Statistics Service (https://www.gks.ru/)

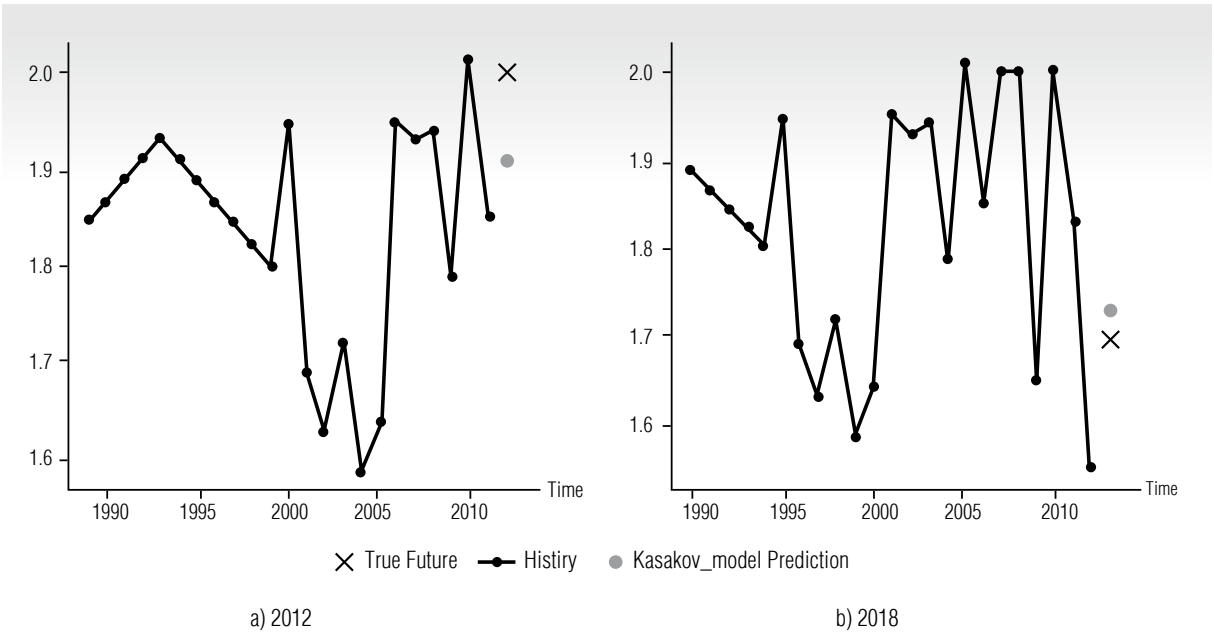


Fig. 10. Determining the value of the target variable using a trained neural network

to other socio-economic systems and processes in terms of solving problems of predictive analytics. The author's approach to the training and use of LSTM networks in socio-economic systems will significantly increase the effectiveness of management decisions. The undoubted advantage of using this technique, in our opinion, is the possibility of early determination of trends in processes, even under conditions of a limited data set.

The proposed approach can become a universal tool for predictive analytics LSTM, since the studied transfer training and domain adaptation techniques in LSTM allowed using the source network trained on synthetic data and predicting the value of the target variable with a high degree of accuracy. The practical significance of the study is to expand the capabilities of the integrated application of simulation methods for building a neural network. At the same time, the approach we developed can be used by state authorities to justify the development parameters of the socio-economic system and allows us to obtain information about its future status. ■

Acknowledgments

The study was carried out with the financial support of the Russian Federal Property Fund in the framework of scientific project No. 18-41-320003 "Mathematical modeling of the socio-economic development of the region in decision support systems using adaptive methods of machine learning and simulation in the face of uncertainty."

References

1. Novoselov A.S., ed. (2014) *Regional and municipal management of socio-economic development in the Siberian Federal district*. Novosibirsk: IEIE SB RAS (in Russian).
2. Kantorovich G.G. (2002) Time series analysis. *Economic Journal*, no 1, pp. 87–110 (in Russian).
3. Anderson T. (1976) *Statistical analysis of time series*. Moscow: Mir (in Russian).
4. Community of IT specialists (2013) *Overview of time series forecasting models*. Available at: https://habr.com/ru/post/180409/ (accessed: 15 March 2020) (in Russian).
5. Pan S.J., Yang Q. (2010) A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no 10, pp. 1345–1359. DOI: 10.1109/TKDE.2009.191.
6. Tsaregorodtsev E.I., Barkalova T.G. (2017) Simulation modeling in forecasting of socio-economic systems. *Herald of TISBI*, no 3, pp. 126–134 (in Russian).
7. Mankaev N.V. (2019) Research and modeling of the process of socio-economic systems management. *Soft Measurements and Computing*, no 1 (14), pp. 21–30 (in Russian).
8. Zvyagin L.S. (2015) Practical methods of modeling economic systems. Proceedings of the *IV International Scientific Conference on Problems of the Modern Economy. Chelyabinsk, 20–23 February 2015*, pp. 14–19 (in Russian).
9. Guo H., Zhu H., Guo Z., Zhang X., Wu X., Su Z. (2009) Domain adaptation with latent semantic association for named entity recognition. Proceedings of the *Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics (HLT-NAACL – 2009). Boulder, Colorado, USA, 31 May – 5 June 2009*, pp. 281–289. DOI: 10.3115/1620754.1620795.
10. Cortes C., Mohri M. (2014) Domain adaptation and sample bias correction theory and algorithm for regression. *Theoretical Computer Science*, no 519, pp. 103–126. DOI: 10.1016/j.tcs.2013.09.027.
11. Gafarov F.M., Galimyanov A.F. (2018) *Artificial neural networks and applications*. Kazan: Kazan University (in Russian).
12. Olah C. (2015) *Understanding LSTM networks. GITHUB blog*. Available at: https://colah.github.io/posts/2015-08-Understanding-LSTMs/ (accessed 12 March 2020).

13. Ganegedara T. (2018) *Stock market predictions with LSTM in Python*. GITHUB blog. Available at: https://www.datacamp.com/community/tutorials/lstm-python-stock-market (accessed 12 March 2020).

14. Kondrashova D.A., Nasyrov R.V. (2019) Comparison of the effectiveness of automatic text classification methods. Proceedings of the *VII All-Russian Scientific Conference on Information Technologies of Intellectual Decision Support. Ufa, 28–30 May 2019*, pp. 146–149 (in Russian).

15. Decree of the Government of the Russian Federation No 2227-R of 8 December 2011. *About the strategy of innovative development of the Russian Federation for the period up to 2020.* Available at: https://www.garant.ru/products/ipo/prime/doc/70006124/#review (accessed: 27 November 2019) (in Russian).

16. *Creating the Monte Carlo experiment*. Available at: https://studme.org/286158/informatika/sozdanie_eksperimenta_monte_karlo (accessed: 27 November 2019) (in Russian).

17. Zvyagin L.S. (2016) Key aspects of complex systems simulation. *Young Scientist*, no 12, pp. 19–23 (in Russian).

18. Drovyannikov V.I., Khaimovich I.N. (2015) Simulation of social cluster management in the AnyLogic system. *Fundamental Study*, no 8–2, pp. 361–366 (in Russian).

19. Yakimov I.M., Kirpichnikov A.P., Isaeva Yu.G., Alyautdinova G.R. (2015) Comparison of simulation results for probabilistic objects in the systems: AnyLogic, Arena, Bizagi modeler, GPSS W. *Bulletin of the Technological University*, vol. 18, no 16, pp. 260–264 (in Russian).

20. Géron A. (2017) *Hands-on machine learning with Scikit-Learn and TensorFlow: concepts, tools, and techniques to build intelligent systems*. Sebastopol, CA: O'Reilly Media.

About the authors

Oleg D. Kazakov
Cand. Sci. (Econ.), Associate Professor;
Head of Department of Information Technology,
Bryansk State Technological University of Engineering,
3, Stanke Dimitrov Avenue, Bryansk 241037, Russia;
E-mail: kod8383@mail.ru;
ORCID: 0000-0001-9665-8138

Olga V. Mikheenko
Cand. Sci. (Econ.);
Associate Professor, Department of Public Administration, Economic and Information Security,
Bryansk State Technological University of Engineering,
3, Stanke Dimitrov Avenue, Bryansk 241037, Russia;
E-mail: miheenkoov@mail.ru;
ORCID: 0000-0003-0917-8406