Forecasting Inflation in Russia Using Neural Networks

Evgeny Pavlov, *New Economic School*

eepavlov@nes.ru

Forecasting Russian inflation is an important practical task. This paper applies two benchmark machine learning models to this task. Although machine learning in general has been an active area of research for the past 20 years, those methods began gaining popularity in the literature on inflation forecasting only recently. In this paper, I employ neural networks and support-vector machines to forecast inflation in Russia. I also apply Shapley decomposition to obtain economic interpretation of inflation forecasts. The performance of these two models is then compared with the performance of more conventional approaches that serve as benchmark forecasts. These are an autoregression and a linear regression with regularisation (a.k.a. ridge regression). My empirical findings suggest that both machine learning models forecast inflation no worse than the conventional benchmarks and that the Shapley decomposition is a suitable framework that yields a meaningful interpretation to the neural network forecast. I conclude that machine learning methods offer a promising tool of inflation forecasting.

Keywords: *inflation forecast, machine learning, ridge regression, neural networks, support-vector machines*

JEL Codes: *C53, E37*

Citation: Pavlov, E. (2020).

Forecasting Inflation in Russia Using Neural Networks. *Russian Journal of Money and Finance*, 79(1), pp. 57–73.

doi: 10.31477/rjmf.202001.57

1. Introduction

Inflation is an important macroeconomic variable that economic agents take into account in their decisions. There are a number of various long-term obligations, such as salaries or loan rates for example, which are generally expressed in nominal prices. Therefore, both households and companies need to forecast inflation accurately. Inflation forecast is also an important input to the monetary policy rule.

One of the key tasks faced by central banks is maintaining price stability, i.e. low and predictable inflation rates. Over the last three decades, in an effort to maintain low and stable inflation, the central banks of many advanced and developing economies have been adopting, in one form or another, the policy of

inflation targeting. The Bank of Russia completed the transition to the inflation targeting policy regime in December 2014. The effectiveness of this policy framework depends heavily on trust with regard to monetary authorities on the part of economic agents. To maintain the trust, central banks regularly announce their forecasts of inflation and other macroeconomic indicators and explain what actions they will take under various possible scenarios. Consequently, the need arose to create a sufficiently reliable and accurate forecasting method. This paper attempts to expand the existing toolkit.

A number of studies show that short-term inflation forecasting (up to two years) is best carried out using simple methods, which are based on the time series of inflation alone (for example, Unobserved Component – Stochastic Volatility Model, UC-SV) and do not require collecting a large number of potential external predictors such as, for instance, the unemployment rate (Stock and Watson, 2008; Faust and Wright, 2013). There are certain restrictions involved in forecasting for Russia that are based on a large number of predictors. Although using a large number of different macroeconomic indicators in forecasting models is potentially justified, the accessibility of data significantly restricts the set of time series considered. Thus, in my research, I use only 10 macroeconomic indicators that reflect the state of the Russian economy and its various sectors. The monthly observations time series restricted by data accessibility may total no more than approximately 200 points.

A high ratio between the number of predictors and observations leads to a high chance of model overfitting. In this context, overfitting implies that the algorithm is fitted to random short-term patterns within a training sample that are not typical for the general population. In other words, a low forecasting error demonstrated by the in-sample model is not preserved if this method is used out-of-sample. This problem is one of the key drawbacks of the models with large number of predictors and, particularly, when such models are used for inflation forecasting.

There are various ways of addressing the problem of overfitting. One of the approaches involves selecting variables based on their theoretical significance. However, this strategy does not necessarily work well over time. Different predictors may have their forecasting value changed depending on various economic issues. Using this method for other countries or time frames may impair the forecasting power. In addition, such a selection of predictors hinders the creation of a universal forecasting method for different forecast horizons. The predictors that have a high forecasting content with regard to prices over two months are likely to differ substantially from those most valuable over two years. Accordingly, this approach is more focused on obtaining good results for a specific sample, rather than on creating a universal algorithm with high accuracy.

The overfitting problem is a serious challenge faced by the machine learning (ML) developers. Over the last few decades, various ML models have

been created: neural networks, decision trees and random forests, support-vector machines (SVM), boosting, etc. These models have become widely used in all fields of data processing and have a high development potential. As far as economics is concerned, it should be noted that ML models can also demonstrate promising results with regard to panel data and time series forecasting. Not surprisingly, with the development of models and growth in their popularity, they started to be applied in the forecasting of macroeconomic indicators. A number of research papers on using ML for inflation forecasting in other countries has already been published (Chakraborty and Joseph, 2017), two of them are dedicated to the application of these methods in Russia (Andreev, 2016; Baybuza, 2018).

Therefore, this paper examines the applicability of ML methods for inflation forecasting in Russia, in particular neural networks. In this research I use a neural network with one hidden layer. To identify the quality of a model, the neural network output is compared with the benchmark econometric standard, autoregression of order one (AR(1)) and the simplest ML method, ridge regression.

It should be noted that due to the high complexity of a model such as a neural network, it is highly criticised because its results cannot be interpreted. The model determines important predictors, but the researcher analysing the results cannot use economic logic to determine the ways these predictors affect the predicted variable. The researcher's inability to tell what is going on inside the applied model also leads to additional criticism of neural networks. As it has been noted in Kleinberg et al. (2015), 'we built them, but we don't understand them.'

To address this problem, I integrate the Shapley decomposition method with ML algorithms (Joseph, 2019). This approach makes it possible to present the results of complex non-linear models in the form customary for a linear regression without losing their significance: the higher the absolute value of the predictor coefficient is, the greater impact it has on the predicted variable. Using and analysing the effectiveness of this interpretation method is also one of the goals of this study.

2. Literature review

To forecast inflation using neural networks, it is crucial to understand not only the ML methods, but also the conventional econometric approaches to inflation forecasting. In their recent survey, Faust and Wright (2013) summarise and compare all the models currently used for inflation forecasting in practice. The authors analyse 17 different models, including autoregression model (AR), UC-SV, random walk, phillips curve, vector autoregression, Bayesian model averaging, etc. Inflation is forecasted in pseudo-real time within the expanding window, starting from the current quarter and on a quarterly basis to a maximum of eight quarters. Root Mean Square Error (RMSE) versus the value of the forecast error in the reference AR(1) model is used as a quality criterion.

The authors have found that the highest forecasting power is demonstrated by the unobserved component model that incorporates information from inflation expectations surveys (primarily the Survey of Professional Forecasters, SPF). Moreover, simple methods such as random walk or AR have proved adequate in inflation forecasting. This paper confirms that complex multi-factor models systematically come short of the classic one-factor models, which are based on the inflation time series alone. Following Faust and Wright (2013), I use AR(1) as a benchmark.

The paper of Chakraborty and Joseph (2017) provides a review of the benchmark ML models applied in addressing certain practical tasks faced by central banks in the field of macroeconomic forecasting. Inflation forecasting in England is studied here as one of the empirical applications of ML. The authors make a forecast with a medium-term horizon of eight quarters. Benchmark macroeconomic indicators (GDP, unemployment rate, money supply, money market rate and other indicators) are used as predicting variables within the window from the start of 1988 until late 2015. All models are trained within the initial window from 1990Q1 to 2004Q4, with further quarterly expansion up until the end of 2013. Mean absolute error (MAE) is used as a measure of quality. The authors also separately consider the value of the forecast error before and after the 2008 crisis. The research paper covers the following models: Bayesian classifier, nearest neighbours model, decision tree and random forest, neural networks, SVM and various clustering methods. AR(1) is used as a baseline model. The authors have found that the combined method of SVM and neural networks delivers superior forecasts before the crisis, while the random forest method yields the best forecast after crisis. These models have not only demonstrated better results as compared to other ML methods, but have also surpassed traditional benchmarks such as AR(1) in terms of forecast quality.

Joseph (2019) applies the Shapley decomposition for conceptual interpretation of macroeconomic forecasts obtained by ML methods. The author considers the Shapley value model and its adaptivity to complex non-linear ML methods. The Shapley value is a linear mapping, which makes it possible to preserve the value of ML regression coefficients. To train the neural networks, SVM and random forest, the author uses quarterly data for the United Kingdom and the United States, which employ the key macroeconomic indicators for 62 and 52 years, respectively. OLS regression with L2 regularisation is used as a benchmark. RMSE serves as a measure of quality. In most cases, ML methods have largely surpassed the OLS reference in terms of the quality of forecasting. Usage of the Shapley decomposition made it possible to ascertain that the most valuable predictors in different models are comparable. The author concludes that ML methods are highly effective for macroeconomic forecasting. Using the Shapley decomposition has turned out to be a useful exercise for simplifying the interpretation of the results of ML regressions.

Szafranek (2017) tries to create an optimum ML method for inflation forecasting in a resource-based economy, taking Poland as an example. This paper is useful for understanding specific issues when applying ML methods for an economy similar to Russia's. The author uses a family of direct ML models with one hidden layer as the basis. At the same time the basic model architecture is used, assuming that the amount of neurons within the hidden layer is distributed in accordance with the Poisson distribution. A large number of single-layer neural networks are trained, after which the results are averaged by means of bagging (bootstrap aggregating). The model is trained based on Polish data, including 188 predictors for the period from January 1999 to December 2016. Performing the exercise, the author makes 72 pseudo-out-of-sample forecasts in a recursive manner starting from January 2011. The forecasts are made for horizons of 1 to 12 months. The quality of forecasts is assessed using the RMSE. The author has found that bootstrap-averaged single-layer neural networks have a better forecast quality than standard models, especially over a long-term horizon.

Andreev (2016) describes the combined method used by the Bank of Russia for inflation forecasting. This algorithm is designed to combine various models, allowing them not to discard preliminary potentially useful predictors. This method includes the following models: random walk, autoregression with linear trend, unobservable component model, classical and Bayesian vector autoregression model, as well as linear regression. After training each of the models, they are aggregated. First of all, the multi-factor models, which were trained using the n number of different samples, are aggregated. After that, the multi-factor and one-factor models are aggregated into a final forecast. This approach makes it possible to use the strengths of different models while engaging all the available predictors. Given that ML methods are also capable of combining various algorithms, creating an effective neural network may become the next stage in the evolution of methods used for forecasting inflation in Russia.

3. Data description and methodology

The data set includes ten macroeconomic time series, including the series of consumer prices. For the purpose of this study, the following main macroeconomic indicators were selected: real GDP, labour productivity (the ratio of real GDP to employment), money supply aggregate M2, real credit growth, unemployment rate, exports, the price of oil in US dollars, real disposable income, money market interest rate, and consumer prices. The consumer price index (CPI), in monthly percentage increments, computed by Rosstat serves as the measure of consumer price inflation. The time series in my sample cover the period from January 2002 to August 2018 (200 observations

altogether). All the data is normalised into monthly growth rates. The physical volume index of GDP computed by Rosstat serves as a proxy for quarterly growth rate of real GDP. Monthly growth rate of real GDP is obtained by linear interpolation. All series are seasonally adjusted, transformed to a stationary form, and standardised.

After seasonal adjustment, the consumer price series was transformed into the first difference of the logarithm to obtain inflation:

$$\pi_t = \log(CPI_t) - \log(CPI_{t-1}).$$

The monthly inflation is shown in Figure 3 in the Appendix. The augmented Dickey–Fuller test confirms that inflation time series is stationary.

Different forecasting models are assessed using the quality functional, which is equal to the RMSE between the real and predicted inflation series:

$$RMSE = \sqrt{\frac{1}{T - T_0 - 1} \sum_{t=T_0}^T (\pi_t - \hat{\pi}_t)^2}.$$

The price level is predicted in pseudo-real time out-of-sample. The need for out-of-sample forecasts is determined by the high vulnerability of ML models to overfitting. For this reason, the model quality is hard to assess without out-of-sample forecasts. Inflation is predicted over the horizons of 1, 2, 6, 12 and 24 months following the most recently available reading of monthly inflation at a pseudo-real moment of time. Models are then compared in terms of their forecast quality separately for each forecast horizon.

Basic AR(1) regression is taken as a benchmark. Accordingly, if the RMSE value for a specific model in a specific month is smaller than the RMSE value of the benchmark, this model is judged to be superior to the benchmark model in terms of the forecast quality, and vice versa.

For a better interpretation of the neural network performance results, the Shapley decomposition is used.

3.1. Shapley decomposition

One of the important tasks in applied macroeconomic forecasting is not only to build a model with high accuracy of forecasts it produces, but also to be able to interpret its predictions. The accuracy of inflation forecasting is not absolute due to a large number of factors influencing future inflation with no hope to explicitly accounting all of them. However, explaining the reasons why the model produced a particular forecast may help researchers find key ways to improve the forecast by adding information about the developments

in those sectors of the economy that the model considers to be the most important.

The Shapley value is principle solution concept in co-operative game theory that yields an optimal allocation of surplus among the players. This allocation is based on the notion that each player's payoff will equal the player's average contribution to the welfare of the total coalition with a certain mechanism for its formation.

For a formal definition of the Shapley value, let us consider a co-operative game with an ordered set of N players. Let K_i denote a subset of first i players in such an ordering. The contribution of the i -th player is defined as $v(K_i) - v(K_{i-1})$, where v is a characteristic function of this cooperative game. The Shapley value in a cooperative game reflects the surplus distribution, where each player receives a mathematical expectation of their contributions to the coalition K_i with equally probable emergence of orderings:

$$\Phi(v)_i = \sum_{K \ni i} \frac{(k-1)!(n-k)!}{n!} (v(K) - v(K \setminus i)),$$

where n is the total number of players, and k is the number of coalition members. The Shapley value satisfies the following properties:

Linearity. The mapping $\Phi(v)$ is a linear operator. That is, for any two games with characteristic functions v and ω and for each α the following is true:

$$\Phi(\alpha(v + \omega)) = \alpha(\Phi(v) + \Phi(\omega)).$$

Symmetry. The player ranking does not affect the payoff received by such player. In other words, if the players are rearranged, the Shapley values are transformed with a similar rearrangement of elements.

Dummy player axiom. The Shapley value for a player not contributing to any coalition, will always be zero.

Efficiency. The Shapley value makes it possible to fully distribute the surplus of the total coalition. In other words, the sum of components $\Phi(v)$ of the Shapley value is equal to $v(N)$.

Strict monotonicity. If the contribution of a player to the welfare of a coalition increases or does not change, his/her Shapley value does not decrease.

Cooperative game theory proves that the surplus allocation among players as defined by the Shapley value is unique and always exists. The Shapley decomposition is a complex exercise that, due to the exponential complexity of the computation, significantly slows down the working of the algorithm. In my study I use a Shapley additive explanation model, following Lundberg and Lee (2017). This model is the most efficient way to compute the Shapley decomposition for its implementation in ML.

The Shapley decomposition is needed for clear interpretation of the results. In the case of linear regression, the interpretation of coefficients is straightforward: the higher the absolute value of the predictor coefficient, the more important it is. This approach, however, may not work for ML methods with a complex structure of parameters.

The procedure I use in this study involves the following steps. First, the values of predictor coefficients are calculated. Each such a coefficient is equivalent to the respective player's contribution, which is required to calculate the Shapley vector. Next, the Shapley value for the i -th variable is calculated using different sub-samples of data in all possible combinations of predictors, after which the absolute values obtained are summed up to obtain the ultimate importance score of the i -th predictor. As a result, the complex and non-obvious coefficients produced by the neural network are presented in linear form, which, as mentioned earlier, is transparent for interpretation: the higher the absolute value of the predictor coefficient is, the greater forecasting value it has with regard to the predicted variable.

4. Models

This section refers to the models I use in this study. First, the traditional AR model is described, which is based only on the previous dynamics of the time series. It is followed by the linear regression model estimated by the regularised OLS (the simplest model in the family of ML methods). Then follows a description of the SVM and neural network – two main ML methods presented in this paper.

4.1. Autoregression (AR)

I use AR model to make an iterated multi-step forecast. That is, to make a forecast for k periods ahead, inflation values are predicted successively between the moments of time t and $t + k$. The iterated multi-step method demonstrates better results than the direct one for predicting the inflation value at $t + k$ (Faust and Wright, 2013). The number of lags p in the $AR(p)$ model is determined using the Bayesian information criterion (BIC). Mathematically, the AR model is formulated as follows:

$$\pi_t = a_0 + \sum_{j=1}^p a_j \pi_{t-j} + \varepsilon_t.$$

4.2. Regularised OLS

ML methods often encounter the problem of overfitting. This is handled by means of regularisation. Regularisation is used to add penalties on coefficients in

the benchmark model. As a result, the algorithm has to find a balance between the accuracy that follows from overfitting and the minimisation of regularisation penalties. OLS regression together with regularisation is a special case of the so-called shrinkage estimator. In this study, a specification with L2 regulariser is used, which is called ridge regression. In mathematical terms, the model is formulated as follows:

$$Q_r(x) = Q(x) + aR(x),$$

where $Q(x)$ is the quality functional, which is the mean squared error, $R(x)$ is a penalty for overfitting, and a is the hyperparameter responsible for the specific weight of regularisation penalties, which is optimally found out-of-sample by means of cross-validation. The ridge regulariser is:

$$R(x) = \sum_{i=1}^q \beta_i^2.$$

Unlike the model with L1 regularisation, it does not set to zero the coefficients for the majority of explanatory variables. This regularization model is chosen to be compared with the neural network model and the Shapley decomposition, which employ the full set of potential predictors.

4.3. Support-vector machines (SVM)

SVMs are a powerful tool for solving classification and regression problems, though it is more frequently used for classification. Generally, a two-class classification problem is solved with the help of a logistic regression. Input data are assessed in accordance with their position with regard to the hyperplane in the feature space and are projected onto interval (0; 1), which is interpreted as a probability of being classified within a certain class.

Using SVM helps address two potential problems that arise from using logistic regression. Firstly, the classification data may be linearly inseparable, that is, the decision boundary may not take the form of a hyperplane. Secondly, the exact separation between the classes may be unknown, even if the exact number of members in classes is known. To address the problem of linear inseparability, data may be projected onto a space with a higher number of dimensions. After that, linear separation becomes feasible using an appropriate transformation. Such data projection is the key element of the SVM algorithm. The second question is how to choose the best separation margin. SVM model chooses the decision boundary in such a way that the vertical distance to the nearest observation is maximised. Thus, only the observations nearest to the decision boundary will be taken when choosing the line which determines the classification across the entire data space. Such observations are called support

vectors, which give the entire model its name. Therefore, the SVM algorithm is a kind of improved logistic regression based on data transformation and geometry of observations.

Figure 1. Graphic interpretation of the two-class classification problem

Source: Chakraborty and Joseph (2017)

The idea of two-class separation is shown in Figure 1. The SVM regression will be addressed at the end of this section. The left panel of Figure 1 shows two types of observations (green and orange points) in a certain feature space. SVM algorithm will try to find a decision boundary leading to the maximum distance from the nearest observations (marked in yellow). Finding the the decision boundary (black line) may be a nontrivial task if the data are linearly inseparable. In this case, the data should be transformed into a new space, where the green and orange points can be separated by a hyperplane, as shown on the right panel of Figure 1. Thus, SVM algorithm first tries to find a new space for the transformed data in which they can be linearly separable, and then determines the margin based on the maximum distance to the nearest observations.

Let us consider the mathematical formulation of this model. Let us suppose that there is a training sample, $(x_1, y_1), \dots, (x_m, y_m)$, $x_i \in \mathbb{R}^n, y_i \in \{-1, 1\}$. SVM model builds the classification function F , which takes the form of $F(x) = \text{sign}(\langle w, x \rangle + b)$, where $\langle \cdot, \cdot \rangle$ is the scalar product operator, w is the normal vector to the separation hyperplane, and b is an auxiliary parameter. The objects in which the F value is equal to 1 are classified into one class, and those in which the F value is equal to -1 in the other class. This specific classification function is chosen on the basis that any hyperplane may be specified in the form of $\langle w, x \rangle + b$, for certain w and b .

Hereinafter, such w and b should be chosen so that the distance to each class is maximised. In this case, the optimisation problem takes the following form:

$$\begin{cases} \arg \min_{w,b} \|w\|^2, \\ y_i(\langle w, x_i \rangle + b) \geq 1, i = 1, \dots, m. \end{cases}$$

This problem is solved by the method of Lagrange multipliers. A regularisation parameter is also introduced in the form of support vector $a_i = C$, which is necessary so that the model does not try to find a clear separation where there is none (if the classes are mixed).

It is obvious that the quality of class separation at the second stage directly depends on how successful the model could transform the data. In case of linear inseparability, we can transform the data by expanding the dimensionality of the feature space. In other words, the data are mapped from a less dimensional space into a more dimensional space in order for the data to become linearly separable. Finding such transformation is a nontrivial problem. However, there is a mathematical solution to this problem. As data enter the Lagrangian in the form of a scalar product, it becomes possible to use the transformation $T(\cdot)$ using the so-called kernel:

$$\mathbb{K}(\hat{x}, \hat{x}') = \langle T(x), T(x') \rangle.$$

The function most commonly used in the ML as a kernel transition function is the basis radial function (Gaussian kernel):

$$\mathbb{K}(\hat{x}, \hat{x}') = \exp(-\gamma \|x - x'\|^2).$$

The SVM can also be used to create a regression based on a continuous series of variable values. The operating principle of regression is similar to the classification model mechanism. In this case, the model has a mechanism similar to linear regression, with the only difference being that now the input data z were transformed by means of kernel transition:

$$y(z) = \sum_{i=1}^m a_i \mathbb{K}(x_i, z) + b.$$

Thus, the model is a linear regression with non-linear data transformation. Like many complex models, SVM are faced with the problem of interpretation. The predictors and the dependent variable have a numerical value, however, using the Shapley value is still reasonable, as with the non-linear kernel, methods that are conventional for linear models will not help find out what is going on inside the model.

4.4. Neural network (NN)

Neural networks are one of the most powerful statistical algorithms in ML. Initially designed for simulating the work of a human brain, neural networks have

become a popular tool in machine learning to solve classification and regression problems. For the purpose of this article, the class of models called multi-layer perceptrons have been chosen from among a wide range of neural networks. I have chosen this type of model due to its good ability to approximate continuous functions. Neural networks may be described as semi-parametric models. On the one hand, model parameters are set using the base network structure. On the other hand, input data are used in the learning process to decide which parameters are significant. This effect does not manifest itself much in this study, but is rather noticeable when using large and complex neural networks on large data arrays.

Figure 2. Graphic interpretation of the neural network structure

Source: Chakraborty and Joseph (2017)

Macroeconomic data enter the neural network for inflation modelling. There is an input layer with a node for each incoming object. The coefficient matrix W_1 (weight matrix) combines the input layer with the hidden layer in the middle of the network. At every node in the hidden layer, the combined signal from the input layer is assessed by the activation function to generate a standardised output signal to the next layer intended for retrieving the final result. This data circulating inside the network are called derived objects. The weight matrix W_2 combines the derived objects with the output layer. The coefficients at the nodes of the activation function are the weights of the weight matrix W . Thus, for the i -th node in the layer k , the incoming object in the activation function will be a derived object of the $(k-1)$ -th layer and the i -th row of coefficients in the weight matrix. NN usually uses the activation function that makes it possible to model a non-linear relation between the predictors and the predicted variable. Such functions may include sigmoid and logarithmic functions. In this study, we use the ReLU (Rectified Linear Unit) activation function. The neural network shown in Figure 2 may be formalised as follows:

$$Y = \text{ReLU}(X \times W_1^T) \times W_2^T \equiv h(X, \beta),$$

where $(X \times W_1^T)$ is the internal product of X properties multiplied by the transposed weight matrix W_1 . With n variables, the input objects, the weight matrix and the output objects X , W_1^T , W_2^T and Y have the dimension $(n \times (n + 1))$, $((n + 1) \times (n + 1))$, $((n + 1) \times 1)$ and $(m \times 1)$; respectively. '+1' in columns reflects the node responsible for the error. Note that if certain inputs in the coefficient matrix are zero, the error node does not pass to this input. Thus, considering 10 explanatory variables, the model will have the parameter $(10 \times 11) + 11 = 11^2 = 121$. Accordingly, for a limited amount of analysed data, the number of layers in the neural network is very restricted.

The activation function is performed by ReLU, which is formally expressed as

$$f(x) = \max(0, x)$$

and is responsible for the threshold transition at point zero. This function works well for computationally intensive operations. Due to the fact that it is expressed as a zero-activation matrix and does not become saturated, the function facilitates better performance of the algorithm. Though this algorithm is subject to the risk that updated weights could cause the neuron to be never activated again, this problem can be avoided by a moderate learning rate.

Despite the numerous advantages of neural networks, their potential drawback is the lack of interpretability. In particular, economic intuition cannot be readily applied to the weight matrix W . Moreover, when working with neural networks in general, different algorithms can give different dimensionality of the weight matrix for the same size of input array. One of the approaches used to solve this problem is to apply the Shapley value to explain the results of ML algorithms.

4.5. Cross-validation and training

In this study, I use cross-validation. Data are randomly broken down into 10 divisions (in the proportion 90% to 10%) for training and testing, respectively. The quality of forecasts is assessed based on the results obtained out-of-sample. Each division is also broken down in the proportion 90% to 10% for calibrating the model hyper-parameters and training, respectively. This procedure is repeated in 15 bootstrap iterations to ensure the stability of results. More bootstrap iterations would result in a higher accuracy of model-based forecasts and less overfitting, but would substantially slow down the algorithm.

5. Results

Table 1 provides the RMSE values for all models in the forecast horizons analysed: 1, 2, 6, 12 and 24 months. Moreover, Table 3 in Appendix shows the values

of the generalisation error (GE), which is equal to the share of the difference between the RMSE values for the training and testing samples and the RMSE value for the testing sample. The data obtained allowed us to draw the following conclusions:

1. Neural networks and SVM have better forecast accuracy compared to benchmark one-factor models.
2. Besides the forecasting horizon of one month, neural networks have a higher forecasting quality in comparison with the AR model, demonstrating a lower average RMSE value by 7% (9% for the forecasting horizons of over one month).
3. Neural networks and SVM have proved insensitive to changes in forecasting horizons of over two months.
4. Neural networks have turned out to be the most prone to overfitting compared to all other models.
5. There is practically no difference between neural networks and SVM in terms of forecast accuracy over a horizon of over two months. However, for the horizon of one month, SVM produce higher-quality forecasts than neural networks and are close to the AR model.
6. The ridge regression model has turned out to be the most sensitive among the ML models to choosing a forecasting horizon. In terms of the quality of forecasts, it has proved to be the worst among the models considered.

Hereinafter is a detailed overview of the results.

Table 1. Forecasting errors

Model	Forecast month				
	1	2	6	12	24
RMSE					
AR (1)	0.2998	0.4500	0.4278	0.4286	0.4369
NN	0.3460	0.3880	0.3980	0.3990	0.4010
SVM	0.3100	0.3910	0.3980	0.3940	0.4070
Ridge	0.3180	0.3910	0.4180	0.4310	0.4230

Note: the best models for each horizon are highlighted in green, and the worst models – in yellow.

The main model used in this paper, the neural network, has demonstrated better results than the benchmark model in the majority of cases. With the exception of switching from a one-month to a two-month horizon, the model is characterised by quite stable forecast error when changing the forecasting horizon. The average forecast error is about 0.4, which corresponds to an increase in the accuracy of the forecast by an average of 9% as compared to the benchmark model.

One reason why non-linear methods have an advantage over linear benchmarks when increasing the forecasting horizon is obvious from the forecast graph over a six-month horizon (see Figure 5 in Appendix). The graph shows a sharp surge in inflation in 2014, which was caused by the depreciation of the

ruble. The AR model memorises and reproduces this surge with a certain lag in the forecasting (equal to six months in this case), which decreases the forecast quality. In contrast, the neural network, which does not use the inflation lag for training, proves to be less susceptible to sharp surges and has therefore an advantage in forecasting if short-term surges are present.

The models analysed do not include the inflation lag. Since they use both the contemporaneous inflation series and the key macroeconomic indicators for forecasting, they turn out to be less sensitive than linear models to changes in the forecasting horizon. Analysing the relation between the predictors and the inflation series, we do not know the exact quantitative effects captured by these variables. A response to possible critics on the need to increase the number of potential predictors for a more accurate description of the state of the economy (which presumably will allow the neural network to identify an overall pattern of interrelations inside the economy) is a small number of available time observations. Such a unfavourable ratio would inevitably lead to the overfitting of the neural network, which is observed already even in the case of 10 predictors. It should be noted that the upsurge reproduced by the neural network in 2014 in a one-year horizon forecast does not prove the model's ability to forecast sudden shocks, and is more likely a consequence of model overfitting. Therefore, if we increase the number of bootstrap samples, this upsurge should disappear. In comparison with other ML models, neural networks turn out to be more susceptible to overfitting. The GE value (see Table 3 in Appendix) when training the neural network is 0.4 in a one-month horizon and 0.212 on average, with some fluctuations for further horizons. This is evidence of significant overfitting of the model, or in other words, its adjustment to peculiarities of the available data. However, this overfitting volume is acceptable for the neural network and does not substantially affect the quality of the model. The other ML models used demonstrate practically no overfitting, but, as the graphs show, they cannot predict local inflation upsurges and downfalls as the neural network does, but instead tend to average the existing series. Thus, although neural networks and SVM preserve the balance between the ability to predict local inflation changes and the absence of overfitting in different ways, both models feature close RMSE values, which indicates a similar level of accuracy for both methods.

From Table 5 in Appendix, which reports the Shapley decomposition for the neural network performance results in various forecasting horizons, it is clear which predictors had the highest impact on the forecast, according to the neural network. For a one-month horizon, the most informative predictors were the GDP, the volume of loans and the inflation rate. Although the contemporaneous inflation has a significant effect on the one-month forecast, it is less than it should be, based on the intuitive understanding of the inflation's nature. Given the neural network's propensity to overfitting for this horizon, we should be careful about the results of the model's performance for this horizon. Later, when the

forecast horizon is increased to up to two months, this problem disappears and the weight increases can be noticed for predictors that reflect the state of key aspects of the Russian economy.

If the forecast horizon is increased to up to six months, the money supply and the interest rate become the most significant variables. The contemporaneous inflation preserves its status as an important variable for forecasting. The forecast chart for six months (see Figure 5 in Appendix) suggests that the neural network predicts some of the local changes quite well. As the degree of overfitting is not too high, the predictors used are most likely to have a decisive influence on the changes in inflation in these local episodes. It is also likely that the remaining local peaks are explained by other predictors not covered by this study.

For the one-year forecast, the most significant predictors were the interest rate, the volume of loans issued and the price of oil. This means that from a neural network perspective, in the case of rather long-term forecasting, the impact of the current inflation evens out and the forecasting power shifts to other indicators that reflect the overall state of the Russian economy and monetary policy.

In the case of a forecast over a two-year horizon, many of the predictors cease to be informative. The interest rate, the unemployment rate and surprisingly, the contemporaneous inflation, have the highest impact on the forecast. Such results can be attributed to the overall complexity of the long-term forecast. Thus, Figure 6 in Appendix shows that the neural network generally strives to follow the mean level, rather than predict sudden peaks. Perhaps using valid forecasts of important macroeconomic indicators as predictors may direct the neural network towards the path of economic development expected by economists. Finding ways to adjust the model to the Russian reality without overfitting is an important and complex challenge for future researchers of this subject matter.

The neural network and SVM demonstrate similar RMSE values. As the SVM practically does not reproduce local changes in inflation, and aims, by its design, to predict the trend, then this method will start losing out to the neural network in future when the number of observations increases. This is due to the fact that the neural network will be able to identify potential key peculiarities of separate episodes and, owing to this, will substantially improve the accuracy of the forecast.

The ridge regression has turned out to be the worst among the ML models, but the quality of the ridge regression and AR models' forecasts has proved to be comparable. The possible problem of overfitting and the model's inability to ignore insignificant variables result in rather poor forecasts.

6. Conclusion

The primary goal of this study is to examine the applicability of neural networks in inflation forecasting in Russia and to compare the results of complex

ML methods with the benchmark forecasting models. The neural network and SVM have generally proved more efficient than one-factor and simple linear models.

The models analysed have demonstrated good results over the forecasting horizon of longer than one month, which makes them a promising instrument for short- and medium-term forecasting.

Using the Shapley decomposition to interpret the forecast of the neural network trained on a macroeconomic data sample has made it possible to identify key predictors in the neural network approach and to determine the scale of their impact on the inflation forecast.

The neural network framework used in the paper is one of the simplest in the family of neural networks. Despite this, it has already shown promising results. Over time, computing power is expected to improve and the amount of training data will increase. This will allow to create more complex models trained on a greater number of variables, which will further increase the usefulness of ML methods to improve the understanding of the quantitative relationships between key macroeconomic indicators.

Appendix is available at
[http://rjmf.econs.online/en;](http://rjmf.econs.online/en;dx.doi.org/10.31477/rjmf.202001.57)
[dx.doi.org/ 10.31477/rjmf.202001.57](http://dx.doi.org/10.31477/rjmf.202001.57)

References

- Andreyev, A.** (2016). *Integrated Inflation Forecasting at the Bank of Russia*. Bank of Russia Series of Economic Research Reports, N 14. [In Russian]. Available at: https://www.cbr.ru/Content/Document/File/16726/wps_14.pdf [accessed on 5 December 2019].
- Baybuza, I.** (2018). *Inflation Forecasting Using Machine Learning Methods*. Russian Journal of Money and Finance, 77(4), pp. 42–59.
- Chakraborty, C. and Joseph, A.** (2017). *Machine Learning at Central Banks*. Bank of England Working Papers, N 674.
- Faust, J. and Wright, J. H.** (2013). Forecasting Inflation. In: Elliott G. and Timmerman A., eds., *Handbook of Economic Forecasting*, Vol. 2. North-Holland: Elsevier, pp. 2–56.
- Joseph, A.** (2019). *Shapley Regressions: A Framework for Statistical Inference on Machine Learning Models*. Bank of England Working Paper, N 784.
- Kleinberg, J., Ludwig, J., Mullainathan, S. and Obermeyer, Z.** (2015). Prediction Policy Problems. *American Economic Review: Papers and Proceedings*, 105(5), pp. 491–495. doi: 10.1257/aer.p20151023
- Lundberg, S. and Lee, S.-I.** (2017). *A Unified Approach to Interpreting Model Predictions*. arXiv preprint arXiv:1705.07874. Available at: <https://arxiv.org/abs/1705.07874> [accessed on 5 December 2019].
- Stock, J. H. and Watson, M. W.** (2008). *Phillips Curve Inflation Forecasts*. NBER Working Paper, N 14322.
- Szafranek, K.** (2017). *Bagged Artificial Neural Networks in Forecasting Inflation: An Extensive Comparison with Current Modelling Frameworks*. NBP Working Paper, N 262.