elements of the series. A neural network with one hidden layer and one output neuron is used. The number of neurons in the hidden layer (90) is three times larger than the size of the time window.

In the output layer, there is 1 neuron acting as a summation unit. A fully connected linear layer (nn.Linear) is used with the ReLU activation function. For regularization, neuron dropout is used with a dropout probability of 0.015. The mean squared error (MSE) loss function (nn.MSELoss()) is used. The optimization method used is torch.optim. Adam with a learning rate of 0.002. Preprocessing of the original series with scaling to the interval [0, 1] is not performed because it introduces additional error into the raw data and leads to additional overhead costs for scaling before and after training.

A short time series of 64 samples is used. The interval between samples is 2s. The duration of the series is 128s. The original series is divided into two datasets — train (70% - 43 samples) and test (30% — 21 samples). Considering that tw = 30, the number of examples for training is 14. Training is conducted for 100 epochs. The batch size is set to 1. The training time for the neural network with the specified architecture on the dataset ˜ 2s. The prediction execution time, with forecasting future 40 samples 1s.

VIII. Description of NN Models Usage Process in the Control System

After obtaining the model for the current state, the main process transfers information about the model to the prediction process. The name of the model file is passed to the predict_file_q queue, and the current utilization data set for forecasting is passed to the predict_list_q queue. The prediction process checks for the presence of data in the specified queues at intervals corresponding to the data collection frequency. After reading the data from the queues, the prediction is executed, and the forecasted data is passed to the model_result_q queue. After receiving the forecast results for the current state, the main process calculates the maximum average utilization (P) across computational modules for the forecast window (Z).

Next, max(R, P) is computed – the maximum between the current utilization value and the maximum forecasted value. This value, together with the code of the current state, the number of computational modules, and the stabilization limit, is used to make a decision about scaling the system. At the same time, if max(R, P) exceeds the addition threshold (A), a decision is made to add resources to the managed system. If max(R, P) is less than the removal threshold (D), a decision is made to remove resources. After the decision is made, a command is sent to the state change block.

IX. Example of the System Under Workload

As an example, Figure 5 illustrates the operation of the system under workload, gradually increasing to 600 users performing various requests to the system over approximately 0.5 hours.

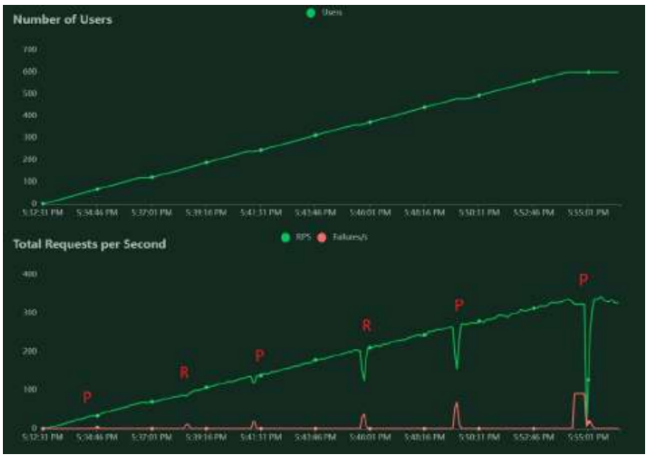

Figure 5. Model System Workload (Number of Users and RPS).

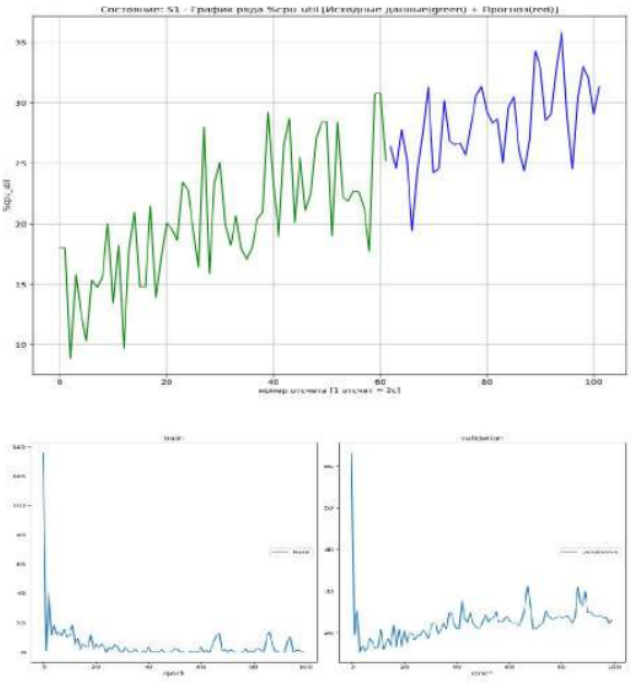

Figure 6. Forecast 1 for state S1.

As a result of the run, the system changed its state 6 times, with 4 changes being proactive and 2 being reactive.

Figure 6 shows an example of utilization forecast for computational modules in state S1 (green color represents the original data, blue color represents the forecast).

Figure 7 illustrates the situation with a reactive state change (sharp peak around the 112th sample).

In this

case, the neural network model did not win, although it predicted the state change threshold exceedance (50%).

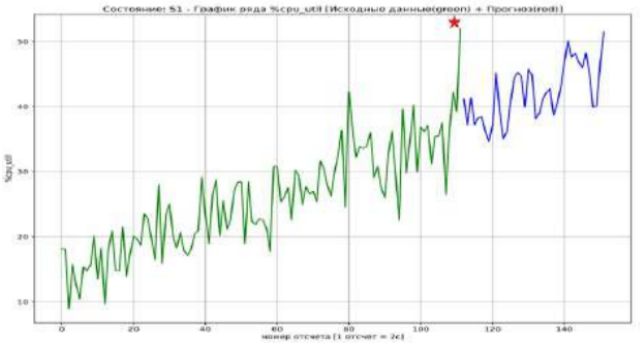

Figure 7. Forecast 51 for state S1.

This situation shows that the complexity of the process increases as the number of users (the number of requests to the system). The neural model does not account for the change in complexity, as the complexity (architecture) of the model itself does not change.

X. Results

The problem related with making real-time decisions in managing the computational resources of a critical IT service under conditions of uncertainty in external load was discussed in the article. As a result of the conducted research, the model system of critical IT service was developed. The architecture of the combined management system are proposed. The technology for real-time decision-making is developed and described, which has been implemented in practice. Experimental researches have been conducted, confirming the workability of the proposed technology.

The original multi-model approach to forecasting problem in case of generating control decisions is proposed, which enhances the adaptive properties and stability of the managed system to external workload. This approach allows to create a library of neural models capable of making forecasts needed parameters for various system states. It becomes possible to further complicate the predictor through the use of ensembles of models. At the same time, a more complex architecture (potentially capable of more accurate predictions) requires increased training time, which reduces the speed of decision making. This situation clearly shows that fast decision-making with preparing a more accurate forecast for a complex signal requires not only an improvement in the algorithm but also more computing resources with high performance.

References

[1] M. Straesser и др. “Why Is It Not Solved Yet?: Challenges for Production-Ready Autoscaling”. В: *ACM/SPEC International Conference on Performance Engineering*. Bejing, China: ACM, 2022, с. 105—115.

[2] N. Khan и др. “Fuzzy Logic applied to System Monitors”. В: *IEEE Access* 9 (2021), с. 56523—56538.

[3] B. M. Nguyen и G. Nguyen. “A Proactive Cloud Scaling Model Based on Fuzzy Time Series and SLA Awareness”. В: *Procedia Computer Science* 108 (2017), с. 365—374.

[4] V. Persico и др. “A Fuzzy Approach Based on Heterogeneous Metrics for Scaling Out Public Clouds”. В: *IEEE Transactions on Parallel and Distributed Systems* 28.8 (2017), с. 2117—2130.

[5] *Docker Compose Overview*. `https://docs.docker.com/compose/`. Accessed: 25.03.2024.

[6] *Kubernetes Cluster*. `https://kubernetes.io/`. Accessed: 25.03.2024.

[7] *Minikube Local Kubernetes Cluster*. `https://github.com/kubernetes/minikube/`. Accessed: 25.03.2024.

[8] *Docker SDK for Python*. `https://github.com/docker/docker-py/`. Accessed: 25.03.2024.

[9] *Echo LabStack High performance extensible minimalist Go web framework*. `https://echo.labstack.com/`. Accessed: 25.03.2024.

[10] *Go, an open-source programming language supported by Google*. `https://go.dev/`. Accessed: 25.03.2024.

[11] *Apache JMeter*. `https://jmeter.apache.org/`. Accessed: 25.03.2024.

[12] *Locust*. `https://locust.io/`. Accessed: 25.03.2024.

[13] *Joblib Python library*. `https://joblib.readthedocs.io/en/latest/parallel.html/`. Accessed: 25.03.2024.

[14] *Multiprocessing Python library*. `https://docs.python.org/3/library/multiprocessing.html/`. Accessed: 25.03.2024.

[15] *Pytorch*. `https://pytorch.org/`. Accessed: 25.03.2024.

[16] A.A. Starovoytov. *Algoritm proaktivnogo upravleniya vychislitelnymi resursami*. 80-ya nauchnaya konferentsiya studentov i aspirantov Belorusskogo gosudarstvennogo universiteta, Minsk, BGU, 2023, 398 s. Accessed: 25.03.2024.

НЕЙРОСЕТЕВАЯ ТЕХНОЛОГИЯ ОПЕРАТИВНОГО УПРАВЛЕНИЯ ИТ СЕРВИСОМ

Краснопрошин В. В., Старовойтов А. А.

В работе иссследуется актуальная прикладная проблема, связанная с созданием систем принятия оперативных решений для управления ресурсами критически важных ИТ сервисов. Неопределенность внешней нагрузки является важным фактором, влияющим на оперативное управление. Для улучшения работы систем управления предлагается подход на основе мультимодельного нейросетевого прогнозирования. Описана модельная система критично ИТ сервиса. Предложена оригинальная технология, структура и архитектура системы управления. Проведены эксперименты, которые подтвердили работоспособность указанной технологии. Received 13.03.2024

Stabilization of Parameters of Technological Operations in the Presence of External Control Actions

Viktor Smorodin *Department of Mathematical Problems of Control and Informatics Francisk Skorina Gomel State University* Gomel, Belarus Email: smorodin@gsu.bys

***Abstract*—A new technique for formalizing probabilistic technological processes when stabilizing the parameters of functioning of technological operations using open semantic technologies for designing intelligent systems is proposed. To stabilize the parameters of technological operations in real time, a procedure for adapting control to variable external control influences is presented based on the neuroregulators algorithms of an intelligent stabilization system that implement control feedback. The principles of development and implementation of intelligent system for stabilizing parameters and a controller for an automated process control system are described.**

Keywords—formalization methodology, neuroregulator algorithms, simulation models, stabilization of controlled parameters, controller

I. Introduction

The operating efficiency of the created automated control systems for real objects largely depends on the quality and adequacy of mathematical models of the research object, which are used at the control system design stage [1]. For this reason, the most significant scientific results in the field of research of the operation of automated production systems and control systems are the development of effective algorithms for adaptive control of automated production based on new methods of neural network modeling of the research object [2] – [4]. The current stage of development of industrial automation means requires ensuring interoperability and semantic compatibility of heterogeneous components of intelligent systems [5].

The paper presents a methodology for synthesizing feedback loops for controlling the technological cycle of automated production based on the use of open semantic technologies for designing intelligent computer systems and algorithms for neural network modeling of control feedback loops.

Vladislav Prokhorenko *Department of Mathematical Problems of Control and Informatics Francisk Skorina Gomel State University* Gomel, Belarus Email: shnysct@proton.me

II. Problems of stabilization of technological cycle parameters

The problem of determining the optimal parameters for controlling technological systems in real time is an important problem of production management in the presence of external control influences during technological operations and random disturbances associated with the design and reliability characteristics of the equipment.

This paper proposes a solution to the problem of stabilizing the parameters of the technological cycle based on the creation of a new generation intelligent computer system capable of stabilizing the parameters of the technological cycle in the presence of external disturbances in real time.

The use of mathematical models of neural networks within the framework of this approach ensures the creation of a new generation of intelligent systems for adapting the control of complex technical complexes, determining optimal control parameters and stabilizing the controlled variables of technological operations in specified ranges of acceptable values depending on external control influences and random disturbances.

The versatility of the proposed approach is determined by a limited set of parameterized procedures that implement algorithms for creating a knowledge base open for expansion based on the ontology of the "probabilistic technological production processes" subject area. The practical and economic significance of the results obtained are determined by the new opportunities provided by the study of existing and design of new complex technological objects.

The construction of a new generation intelligent computer system involves the following development stages:

- methods for formalizing the technological production process based on the use of the ontology of the subject area "technological production processes with probabilistic characteristics";