

Article

Machine Learning Prediction Techniques in the Optimization of Diagnostic Laboratories' Network Operations

Krzysztof Regulski ^{1,*}, Andrzej Opaliński ¹, Jakub Swadźba ², Piotr Sitkowski ², Paweł Wąsowicz ² and Agnieszka Kwietniewska-Śmietana ²

¹ Department of Applied Computer Science and Modelling, AGH University of Krakow, Al. Mickiewicza 30, 30-059 Kraków, Poland; andrzej.opalinski@agh.edu.pl

² Diagnostyka S.A., Życzkowskiego 16, 31-864 Kraków, Poland

* Correspondence: regulski@agh.edu.pl

Abstract: The article presents an outline of the concept of a prototype system allowing for the optimization of inventory management in a diagnostic laboratory on the basis of patients results. The effectiveness of laboratory diagnostics depends largely on the appropriate management of resources and the quality of tests. A functional quality management system is an integral element of every diagnostic laboratory, ensuring reliability and appropriate work standards. This system includes maintaining correct and reliable analytical test results as well as the optimal use of the laboratory equipment's processing capacity and the appropriate organization of the supply chain—both analytical material and reagents. It is extremely important to avoid situations in which tests cannot be performed due to a lack of reagents, the overloading of analyzers, or improper calibration. Therefore, the accurate prediction of the number of orders is crucial to optimize the laboratory's operations, both in the short term—for the next few hours and minutes—and in the longer term, even monthly, which will allow for the appropriate planning of reagent stock. As part of the research presented in this article, machine learning methods were used to implement the above functionalities, which allowed for the development of a prototype of a laboratory optimization system using patient test results as a basis.

Keywords: machine learning; optimization; quality control; laboratory diagnostics



Citation: Regulski, K.; Opaliński, A.; Swadźba, J.; Sitkowski, P.; Wąsowicz, P.; Kwietniewska-Śmietana, A.

Machine Learning Prediction Techniques in the Optimization of Diagnostic Laboratories' Network Operations. *Appl. Sci.* **2024**, *14*, 2429. <https://doi.org/10.3390/app14062429>

Academic Editor: Frede Blaabjerg

Received: 19 January 2024

Revised: 8 March 2024

Accepted: 11 March 2024

Published: 13 March 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The analyzed example of Diagnostyka S.A. diagnostics network conducts over 80 million tests annually, serving more than 11 million patients per year and processing over 60 thousand samples per day. Managing such a large volume of data poses significant infrastructural challenges, encompassing various Big Data aspects. The research presented herein aimed to develop predictive methods for both the overall number of tests and the test outcomes for individual patients based on their previous results. This prediction allows for the optimization of laboratory work on multiple levels. One of these levels involves optimizing inventory management. By knowing the projected number of tests, it is possible to precisely adjust the inventory of reagents and consumable materials, such as tubes, to minimize storage costs while maintaining workflow efficiency in analyzers and preventing downtime due to shortages of consumables and reagents. The second level of optimization involves quality control. Currently, quality control is performed at least once a day for a specific parameter, sometimes more frequently. If the control measurement indicates any deviations, recalibration of the device is necessary. Anticipating the need for recalibration by predicting the meta-characteristics of a parameter in patient tests could reduce the number of control measurements required. This would decrease the consumption of control materials and analyzer runtime necessary for statistical assessments of both preliminary and actual control measurements.

The diagnostic process described in Figure 1 processes data acquired from one of the largest companies in Poland that carries out laboratory tests on biological materials obtained in their own facilities and supplied by customers such as hospitals, health clinics, and collection points. The research was conducted using the 10 most frequently performed types of tests, which accounted for a total of 83% of the tests, resulting in a set of 140 million tests over a three-year period.

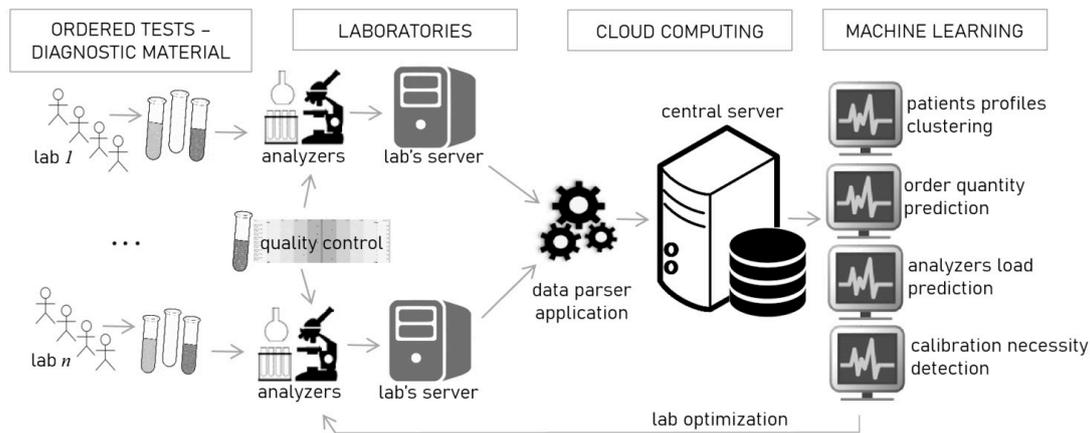


Figure 1. Schematic diagram of the research: data flow in the system and main tasks in the field of machine learning.

In quality control, methods that involve assessing the quality of laboratory test results based on results obtained from control materials provided by reagent manufacturers have been used for decades. With current production precision, the standard deviations calculated based on control materials are minimal. The necessity for quality control is assessed in relation to exceeding standards calculated based on these deviations, resulting in a very rigorous quality maintenance regime. This ensures a very precise testing process. However, as a consequence of such procedures, many device calibrations and control measurements are performed unnecessarily because the device does not actually exhibit any real deviations. An alternative is the PBRTQC (Patient-Based Real-Time Quality Control) approach, discussed more comprehensively in Section 3.4, essentially involving conducting and maintaining quality control not based on artificial control materials but on indicators derived from actual patient test results and the distribution characteristics obtained from these results.

PBRTQC, emerging as an alternative or complementary method to quality control, has attracted considerable interest, and there are numerous strategies that interested users can explore for application in their own laboratories. Nonetheless, there are lingering concerns regarding its effectiveness and practical clinical contexts. Its ability to be applied has been brought into question. According to Tony Badrick [1], there is minimal software assistance available, and significant time and statistical expertise are necessary to establish a PBRTQC program. Official guidance documents outlining the pros and cons of the methods are lacking. However, the Clinical and Laboratory Standards Institute guidelines on risk-based QC stand out as an exception, suggesting that the monitoring of aggregated patient results as a QC approach is unaffected by matrix effects.

Current research challenges in the context of patient-based quality control methods include issues related to developing specific methods and guidelines to establish norms and protocols regarding optimal quality indicators for monitoring and analyzing patient data in real time [2,3] and the integration of data from various medical and laboratory systems for effective quality monitoring [3], as well as the implementation of information systems—the development of IT tools enabling integrated analyses of patient data with laboratory systems and prompt responses to detected irregularities [2,4].

The present study aims to demonstrate that such an approach can indeed be applied in practice. However, we go further by expanding the functionality of methods based on patient study data. The innovativeness of the proposed solution lies in utilizing these data not only for quality control but also for optimizing reagent management by forecasting demand, optimizing device operating time, and enabling the possibility of profiling patients based on their results. The subject of this analysis is a solution regarding the use of machine learning in the quality control of medical analyzers and laboratory workflow management elements using knowledge obtained from order data, as well as the solutions found in the literature regarding the use of various machine learning models in calibrating PBRTQC methods. They also propose different monitoring indicators. However, they typically focus on individual studies, single-analyte parameters [2–5]. In the proposed solution, the use of data from analytical studies can significantly influence the broader functioning of the laboratory in various aspects.

Patient data that are dispersed throughout the national network of laboratories is stored on distributed servers as part of the IT infrastructure of analytical laboratories. To make these data available for processing as part of the presented work, it was necessary to unify and aggregate the data into one common, unified database model of data and metadata. The models developed in this work pertain to the universal, unified, and aggregated metadata model describing the results of laboratory tests conducted on patients. In addition, structures were developed to store the operating characteristics of the laboratories, including the biochemical analyzers, their load, efficiency, and the entire quality control process of the tests performed on these analyzers. This also enabled the analysis of these aspects of laboratories' operations.

As part of the described research, IT development work resulted in a central server that integrated over 140 million tests conducted on approximately 11 million patients over a three-year period. The data also include information on the characteristics of the tests performed, device load, and quality control parameters for the laboratory process. The data collected in this manner form the basis for further analysis, which is the subject of the work presented here.

The issue of data diversity from different laboratories was solved at the level of data integration. Data from each laboratory required a separate data parser and the transformation of its schema into a common schema developed for analysis on a central server, where the sources were integrated (Figure 1). However, if we were to look at the variability of data in terms of distribution characteristics, this problem is basically solved by the size of data streams. The laboratories that we integrated are located in large cities, each of them is itself a large source of highly diversified data (both hospitals, patients from outpatient clinics and clinics, and individual patients). Each laboratory belongs to the same diagnostic network, which means that it is subject to the same corporate regulations, has the same standards and procedures, and often has devices of the same class and type. Therefore, the common problem of maintaining consistent test conditions across facilities due to variations in sample collection and analysis times does not occur here. In each case, these are demographically diverse data covering the entire spectrum of the patient cohort. Bias is thus negligible. However, the analyses were performed on full, integrated sets, where a single patient or parameter was compared against a nationwide cross-section. It was different only in the case of assessing the load of a single device, where the data came from only one location. But the comparative background was historical data from the same analyzer, and the problem of location did not play a role.

2. Materials and Methods

The initial step in this process involved creating a universal and unified metadata model that allows for the collection of key characteristics of data, including numerical, textual, and qualitative information, for selected laboratory tests. The resulting universal data model was developed using the Oracle relational database engine. Dedicated external data parsers were implemented to migrate selected data concerning a specific type of

research from a specific period of time to a previously developed universal data aggregation database model.

The key steps in implementing these tasks were to ensure the following:

- Data synchronization between laboratories—the decentralized database system resulted in data inconsistency in the context of handling single-patient data in different analytical laboratories. This issue had to be handled while still considering data anonymization.
- The anonymization of data processed in the system—in order to avoid processing the sensitive personal data of patients, the process of data integration had to be anonymized. This was achieved by storing only the patient's date of birth and gender in the universal, integrated data model. The remaining personal data are anonymized and masked with an appropriate hash function to a unique key, making it impossible to read the patient's sensitive personal data.
- Consistency of data between laboratories—in order to store data in a universal model, it was necessary to normalize the data features and characteristic parameters, which describe the diagnostic process, test results, and qualitative and quantitative parameters. The values initially used were significantly different in various laboratories and diagnostic analyzers.
- Conducting a multi-day data import and integration process while maintaining full availability of source systems in which the diagnostic process was carried out in real time, resulting in the appearance of new data and updating the source databases of research results.

The data aggregated as part of the research are stored on a central server. The server is implemented on an HPC server with a Supermicro H11SSL motherboard (Super Micro Computer, Inc., San Jose, CA, USA) and an AMD EPYC 7401P processor (Advanced Micro Devices, Inc., AMD, Santa Clara, CA, USA). It has 128 GB of RAM memory and SSD drives with a total capacity of 12 TB. The operating system used is Windows Server 2019, and the database engine is Oracle.

As part of the research, Figure 1 presents a schematic representation of the four basic stages of the work conducted in the group of machine learning tasks.

- (1) The analysis of anonymized data from the blood test results of patients, the segmentation of sequence results for each patient, and the creation of patient profiles based on these results. To achieve this, we employed methods for segmenting sequential data, including unsupervised learning methods such as Kohonen networks (self-organizing maps, SOMs), which enable correlational learning to search for objects similar to each other in terms of sequence characteristics. Sequence analysis tools, including PrefixSpan 2.4.0 and RNN (LSTM) (from Keras framework), were used for modeling. The models were trained on datasets of measurement sequences for patients. The test set comprised 49,200 records. After filtering out only patients who underwent tests at least five times, the dataset size was reduced to approximately 4100 series. Validation was conducted on a set consisting of 1906 series with lengths ranging from 5 to 10 measurements.
- (2) The obtained patient profiles were utilized to personalize the prediction models (sub-models) for the number of test orders. By understanding the characteristics of patient groups, we can determine the likelihood of whether patients from a particular group will repeat tests (those under treatment) or have completed therapy. Predicting orders from a group that does not repeat tests (first tests) requires a different type of modeling because we lack knowledge about previous results. Therefore, we can only rely on time series analysis and autocorrelation. Exponential smoothing models were utilized to analyze time series data related to the overall number of orders. This enabled the development of prediction models for the global number of orders in patient groups, with an accuracy range of 91–99% Determination Coefficient (R^2), depending on the parameter. The analysis of prediction accuracy was performed on a dataset concerning 11 analytes, totaling 1,388,845 orders from patients over six months (calculated from a

- period of at least 3 months from the two previous years from the same time periods). The means and standard deviations of these samples were compared.
- (3) By having datasets of results for a specific test parameter, meta-characteristics can be observed based on all the results. Changes in their distribution may indicate disturbances related to the calibration of devices such as analyzers. This approach to quality control, based on measurement data, allows for the effective detection of the necessary calibration moment and constitutes another machine learning task. Multi-layer perceptrons (MLPs), which are a type of neural network, were used and found to be effective. For the analysis, 16.5 thousand measurement results and 13 calibration points were utilized. As a result of the study, 9 parameters which, in a combined model, enable the assessment of measurement effectiveness were developed.
 - (4) The last task involved predicting analyzer overload. Diagnostic laboratories have a cyclic daily pattern of work, with the testing of samples from hospital patients and medical units continuing throughout the day. However, the largest quantity of samples is collected during the peak period, when material is collected from partners serving as blood collection points. This peak is associated with the transfer of material to couriers, which can occur once or twice within an hour of operation. If all the material flows into the device simultaneously, the queue may exceed the analyzer's capacity. By anticipating the moment of overflow, we can direct the material to other less burdened devices to avoid queuing and increase overall daily testing efficiency. Logistic regression and CART classification trees were used to create effective models for predicting maximum load. In the case of analyzer overload, data pertained to the six most burdened laboratory centers and covered the period of 2020 and 2021. Through aggregating information on the central server, data from a total of 434 analyzers processing data in these laboratory centers were obtained. Taking into account their cumulative load, this number can be reduced to 201 analyzers, assuming that optimization will only affect analyzers performing an average of at least 21 tests per day, and similarly, 88 analyzers performing at least 100 tests per day and 10 analyzers performing at least 1000 tests per day. The group of the most burdened analyzers was considered (as the benefit from their optimization is the greatest). For testing, to avoid systematic errors associated with seasonal load, a sample of 30 daily runs was randomly selected from 5 random analyzers from the group of the most burdened (data concerning one day of work for 5 random analyzers) from a period of at least 3 months of work (the years 2020 and 2021 were considered). As a control group, 30 random daily runs were selected from the two previous years and the same time period (appropriate 3 months, appropriate 5 random analyzers). On average, this group of analyzers performs approximately 23,000 tests per day.

3. Results

3.1. Patient Profiling—Clustering

The segmentation of patient profiles was established on the premise that healthy patients undergoing only follow-up tests typically do not repeat those tests within the same year. The research findings do not allow for predicting future orders in this group. However, there exists a sizable subset of patients (approximately 10%) who undergo tests more than 10 times, indicating progress in their therapy. Some patients repeat tests until reaching a specific expected level of the parameter, signifying recovery or another condition, such as pregnancy in the case of the bHCG test. In these cases, it is possible to deduce from the test results whether they will return for additional tests and within what time frame.

The initial step involved demonstrating the statistical differences between patient groups to facilitate further conclusions. Self-organizing maps (SOMs) were utilized for this purpose [6]. SOMs are Kohonen neural networks designed for unsupervised learning, particularly useful for clustering sequential data like time sequences or time series. They can cluster and analyze patterns within these data. The process of clustering sequence data using SOMs involves training a SOM map on the data to represent the data's structure and

relationships [7]. SOMs can identify similar patterns in temporal sequences and group them based on their similarity characteristics. For instance, when analyzing temporal sequences in patient data, SOMs can assist in identifying similar study patterns.

The implementation of the models was carried out in Python using the MiniSom library and various tools for data analysis and machine learning, such as numpy, pandas, keras, tensorflow, etc. SOM was used for clustering sequence data of patient examinations and then presenting the results in the form of a cluster map. Cluster and sequence data were then used to train LSTM networks for future patient classification into specific profiles. The data underwent preprocessing, including normalization and padding/extending to a constant length. In the next stage, appropriate parameters for the self-organizing map were selected, and then the map was trained on the data. The implementation in the MiniSom library does not require the predefinition of layers or neurons. A SOM is a one-dimensional layer containing neurons, each of which is connected to all inputs, thus adapting to input vectors and their lengths.

The outcome of this process was a mapping of temporal sequences based on their similarities, enabling the visualization and identification of clusters or patterns, which proved valuable for analyzing and understanding the structure of the sequence data (see Figure 2) [8].

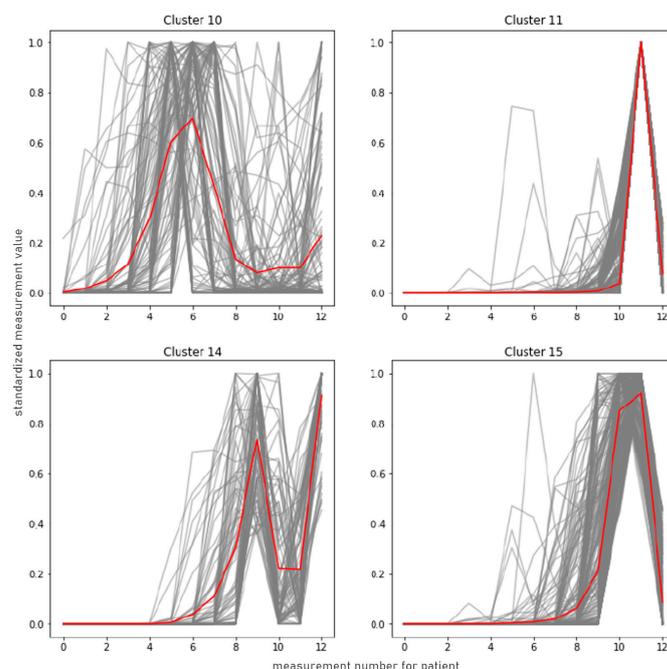


Figure 2. Four selected clusters of SOM clustering results.

The research results presented in Figure 2, characteristic of the four selected clusters, illustrates how typical variations in the outcomes of a given study may manifest in individual patients. We have a normalized value of the bHCG test on the Y axis, and the number of tests in the series is placed on the X axis. In individuals who constantly monitor the level of a certain parameter (in this case, betaHCG), this level may undergo characteristic fluctuations, which may indicate a disorder (in this case, desired pregnancy is often the case), effectiveness or ineffectiveness of therapy, etc. Depending on changes in the parameter value, further steps in therapy can be predicted and, consequently, so can specific patient behavior regarding further tests (whether a repeat test should be expected or not).

Clustering was performed using self-organizing maps, and the number of clusters was selected based on the Silhouette index. The algorithm takes a list of numerical sequences, map size, learning rate, sigma, and network size as its input and produces a list of clusters

with sequences assigned to them as its output. Silhouette Index Analysis is a method used to interpret and validate consistency across data clusters. The indicator value measures the similarity of an object to its own cluster compared to other clusters. It can be used to study the separation distances between resulting clusters. The ratio is calculated by averaging the distance between clusters (a) and the distance to the nearest cluster (b) for each sample.

Additionally, the PrefixSpan sequence pattern discovery method was employed [9]. This method aims to identify the most frequently occurring sequences of events based on a list of sequences and a minimum support value that determines their frequency. The algorithm also identifies the longest sequences, assuming a minimum number of repetitions. The algorithm also identifies the longest sequences, assuming a minimum number of repetitions. The results are sorted based on the specified criteria. The method presented produces a set of sequence patterns. Each tuple in the set contains two values: the support of the pattern and the recognized sequence, which is a list of cyclically repeated values. The data were discretized (qualitized) using the KBinsDiscretizer method [10] from the sklearn.preprocessing library to ensure they could be used with the PrefixSpan method. Discretization is the process of transforming continuous data into discrete values for numerical processing. The method used grouped variable values into countable containers and assigned each a unique integer while maintaining the ordinal relationship between them.

This research study proposes a third algorithm for classifying sequences into individual categories using LSTM recurrent neural networks [11]. The recurrent network is trained using the sequence and the SOM clustering result from the previous method. Test data are fed to the trained network to determine the correct operation of the algorithm. The resulting classifier can be used to classify patient data. A recurrent neural network is a type of artificial neural network that creates a directed or undirected graph along a time sequence [12]. Its use demonstrates the dynamic behavior of data over time. Unlike feedforward neural networks, recurrent neural networks use their internal state (memory) to analyze sequences of variable-length input data. LSTM (Long Short-Term Memory) is a type of recurrent neural network that is designed to work with sequential data such as text, time series, audio, or action sequences. It has the ability to remember long-term dependencies in data [13].

LSTM is distinguished by its capacity to retain information for longer periods than standard RNNs, thereby eliminating the vanishing gradient problem that occurs in classic RNNs. This is accomplished through the use of special structures known as 'gates' that determine which information should be retained and which should be discarded during sequence processing [14].

The data are divided into training and testing sets (80:20) to train the model and evaluate its performance, as well as its accuracy on data it has not seen before. The utilized recurrent neural network model consists of the following layers:

- Input layer: The number of features in the input data depends on the lengths of series (sequences).
- Hidden layers:
- LSTM layer: 100 neurons—A type of recurrent network that utilizes special units in addition to standard units containing memory. Thanks to these special cells, they can store information in memory for a long time, allowing them to learn long-term dependencies.
 - Dropout: 50% dropout rate. No neurons; it is a regularization layer. It is a layer that randomly sets input units to 0 with a certain frequency at each step during training, which helps prevent overfitting.
 - Dense layer (hidden layer): 100 neurons with ReLU activation function.
 - Output layer: Dense layer: The number of outputs, which is equal to the number of clusters (25 in this case) with softmax activation function.

The training process involves the use of the categorical_crossentropy loss function, which is typically used in multi-cluster classification tasks where a sequence can belong

to one of several categories and the model must determine the appropriate category. The recurrent network was trained multiple times, and the average results are presented in Figure 3. A graph of loss and correctness during specific epochs of data training is also provided for the best result. The results of training and testing recurrent neural networks for recognizing sequences belonging to particular categories were 91.79%. The graphs presented in Figure 3 illustrate the training process by showing the accuracy and loss values in individual epochs.

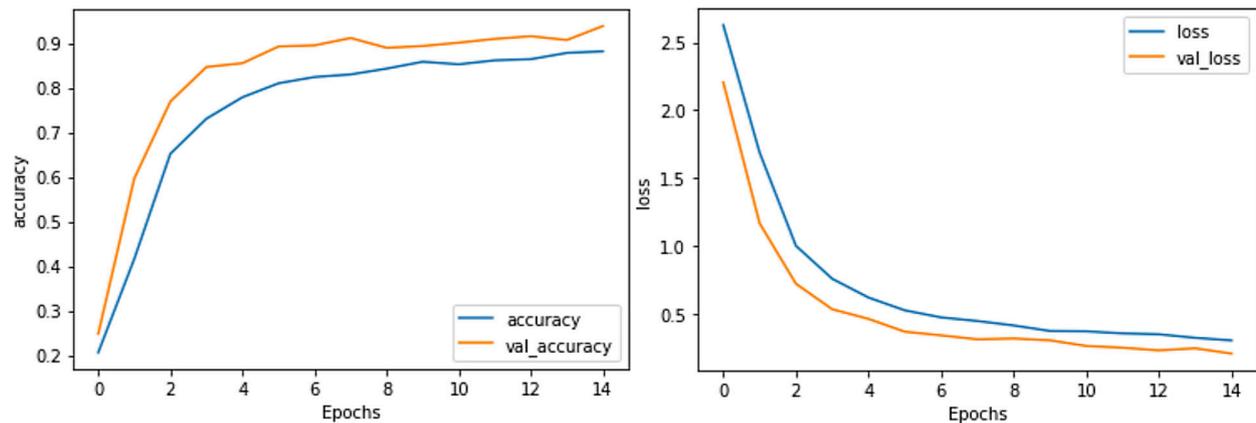


Figure 3. Graphs of accuracy and losses in training the recurrent neural network model.

Accuracy and loss graphs are common tools used in training recurrent neural networks (RNNs) to monitor the performance and progress of the model during training. Accuracy refers to the proportion of correctly classified instances among the total instances. In the context of RNNs, an accuracy graph shows how well the model is performing in terms of making correct predictions. Accuracy is plotted against the number of training epochs (iterations through the dataset). At the beginning of training, accuracy is low as the model is still learning patterns in the data. As training progresses, accuracy increases as the model learns to make better predictions. Accuracy graphs provide insights into whether the model is learning and improving over time. A consistently increasing accuracy curve indicates that the model is learning effectively, while fluctuations or a plateau may suggest issues such as overfitting or underfitting.

Loss, also known as error, measures how far the predicted values are from the actual values. A loss graph shows the value of the loss function (categorical cross-entropy) over the training epochs. The goal during training is to minimize the loss, that is, to make the predicted values as close as possible to the actual values. Similar to accuracy, loss is plotted against the number of training epochs. At the beginning of training, the loss is high as the model makes random predictions. As training progresses, the loss decreases as the model adjusts its parameters to make better predictions. Loss graphs help monitor the convergence of the model during training. A decreasing loss curve indicates that the model is learning and adjusting its parameters effectively.

Monitoring both graphs during training helps to assess the performance and convergence of the recurrent neural network.

3.2. Prediction of the Number of Orders—Exponential Smoothing

As part of the research, a universal model was selected to suit diverse time series of various types of research and provide precise data on the number of orders aggregated monthly. The chosen forecasting model is based on exponential smoothing, which uses a weighted moving average to reduce variance and predict future values of the series. The exponential methods employed include Brown's Model, Holt's Linear Model, and Winters' Model, all time series forecasting techniques.

The Winters' Exponential Smoothing Model is an extension of the classic exponential smoothing model and takes into account seasonality and trends in the data [15].

- Level Component: Represents the baseline level of data over a given period of time. In Winter's Exponential Smoothing Model, it is updated according to the last observation and takes into account current level changes.
- Seasonal Component: Represents cyclical changes in data, such as months of the year, days of the week, etc. It is accounted for in Winter's model by including a seasonal factor modifying the forecasts.
- Trend Component: Represents a long-term upward or downward trend in the data. It is included to predict future trend changes.

Exponential smoothing is a forecasting technique that assigns exponentially decreasing weights to historical observations. This means that the future value is not only dependent on the last observed value but on the entire series of values. The influence of older values is smaller than that of newer values. It is important to note that this technique assumes that the future value depends on the entire series of values.

Exponential smoothing models are characterized by four parameters (α , β , γ , ϕ), and various initialization methods are considered. The β parameter controls the trend, while the ϕ parameter controls the strength of extinguishing the trend. The γ value is responsible for seasonality in the model. The main challenge in the algorithm is selecting the appropriate parameters to achieve the most accurate forecasts. Furthermore, the model allows for an independent determination of the nature of each component—trend, seasonality, and residuals—detecting whether each component is either additive or multiplicative. It is also possible to assume that a component does not appear in the series, particularly for trend and seasonality. To shorten the notation, we use the standard ETS designation for exponential smoothing models. The model components are represented by the letters E for error, T for trend, and S for seasonality. The appropriate symbols for each component type are then inserted: A for additive, M for multiplicative, and N for none (only for trend and seasonality). In the case of a damped trend, the letter 'd' is added. For instance, a model refers to a set of models with additive errors, a multiplicative damped trend, and no seasonality.

Subsequent modifications and extensions of this algorithm gave it the form currently popularly used: AAA ETS, i.e., AAA = Additive Error, Additive Trend, and Additive Seasonality; ETS = Exponential Triple Smoothing algorithm [16].

The research scenario stated that the ten parameters with the highest reagent consumption (i.e., the largest number of tests) would be selected for testing. A list of the ten most frequently performed tests was compiled, taking into account that some tests do not require reagents (such as blood count tests or urinalysis). Although bHCG was only ranked 43rd, it was included in the list due to the repeated tests on this hormone and its prediction being used as a reference value.

Three separate forecast sub-models were developed for each of the tests mentioned in Table 1, one for each patient profile. Furthermore, it was noted that order values on weekends seldom exceed 5%. As a result, separate models were created for weekend forecasts. In total, 33 time series simulations were conducted, each spanning 21 months (with a three-month delay for the initial forecast). The total number of regular forecasts was 693, with an additional 33 weekend forecasts. Figure 4 provides a few examples of individual forecasts.

Table 1. Number of tests on individual parameters.

| No. | TEST_CODE | NAME | Number of Tests |
|-----|-----------|-----------|-----------------|
| 1 | 100 | TSH | 1,364,261 |
| 2 | 20 | ALT | 858,255 |
| 3 | 61 | CRP | 763,815 |
| 4 | 21 | AST | 581,132 |
| 5 | 101 | FT4 | 483,112 |
| 6 | 13 | Potassium | 465,509 |
| 7 | 40 | Iron | 335,365 |
| 8 | 12 | Sodium | 287,343 |
| 9 | 114 | Prolactin | 102,062 |
| 10 | 112 | Estradiol | 72,785 |
| ... | ... | ... | ... |
| 43 | 116 | Beta-HCG | 55,689 |

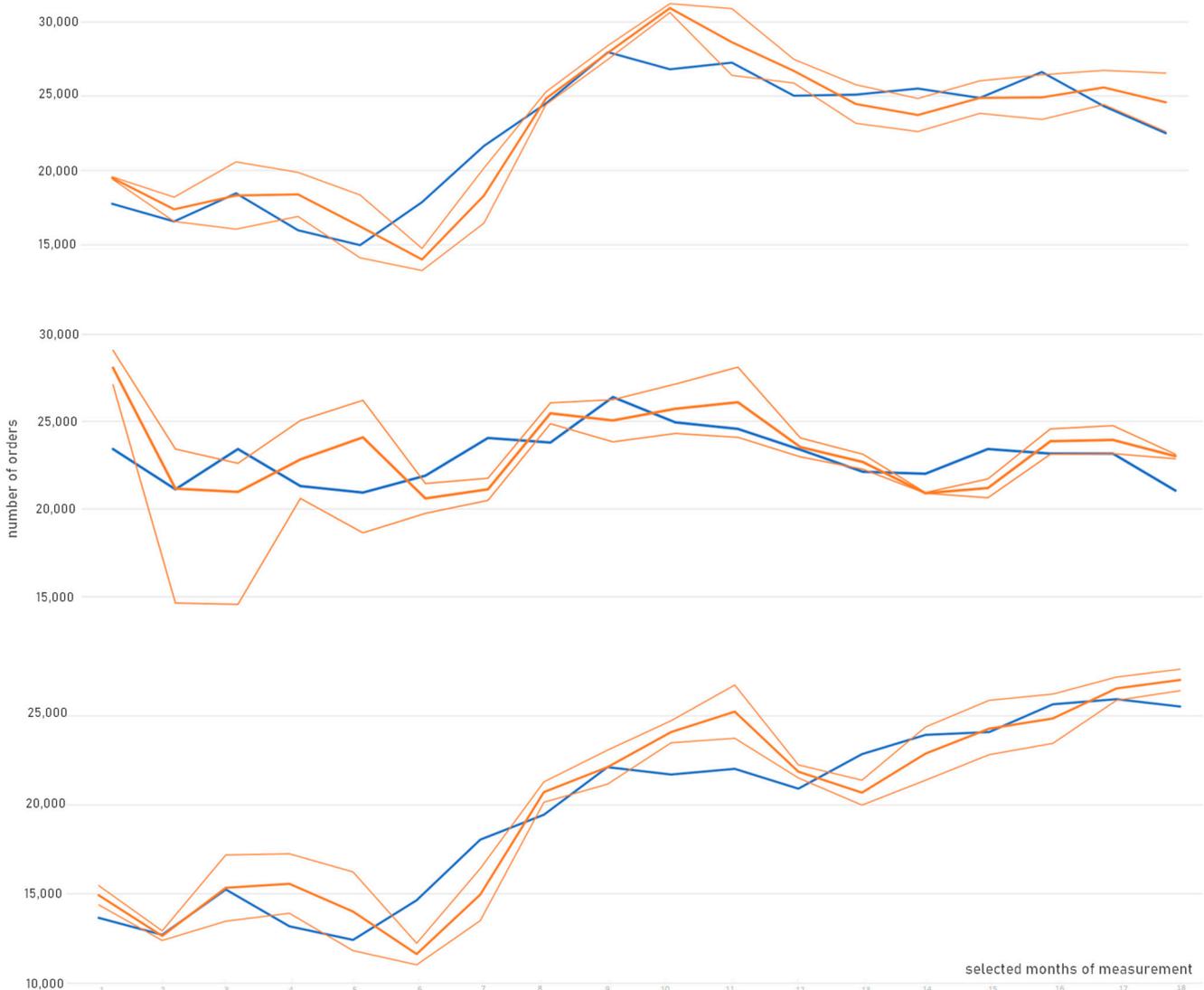


Figure 4. A few examples of individual time series simulations. The blue line indicates the actual course, while the bold orange line indicates the forecast, while the thinner orange lines indicate the confidence interval.

A prediction plot with confidence intervals overlaid on plotted historical data from the same period is shown in Figure 4. The forecasts were calculated based on data from two

three-month periods from each of the three previous years; therefore, the timeline (x-axis) cannot be regarded as a continuous range but rather as individual measurement points. Each of the plots depicts the trend and prediction for a different examined parameter. It can be observed that at some points, the prediction lags by one time step from the historical data; however, the confidence intervals cover the variability quite well, especially considering that the studied period includes the time of the pandemic. It is also worth noting that the period during which the models were examined coincided with the pandemic, which disrupted the blood research market, affecting its dynamics.

Table 2 presents the results of forecasts for 11 factors, including the 10 most commonly performed plus bHCG, over a six-month period. The forecasts were calculated based on data from two three-month periods from each of the three previous years during the same periods of the year. The table compares the averages and standard deviations of these samples.

Table 2. Results regarding the number of forecasts and the corresponding parameters.

| No. of Tests (Parameters) | | 11 |
|---------------------------|--------------------------------|-----------|
| RSS | Residual Sum of Squares | 4,711,975 |
| MAE | Mean Absolute Error | 1559 |
| RMSE | Relative Mean Square Error | 0.01 |
| d | Relative Average Deviation | 0.08 |
| r | Correlation Coefficient | 0.99 |
| R ² | Determination Coefficient | 0.98 |
| MAPE | Mean Absolute Percentage Error | 2.16% |

The Mean Absolute Percentage Error (MAPE) is the percentage error calculated separately for each parameter within a given set of forecasts and then averaged. The MAPE value of 2.16% indicates the precision of the models. Figure 5 shows a scatter plot of the model predictions compared to the observed order quantities.

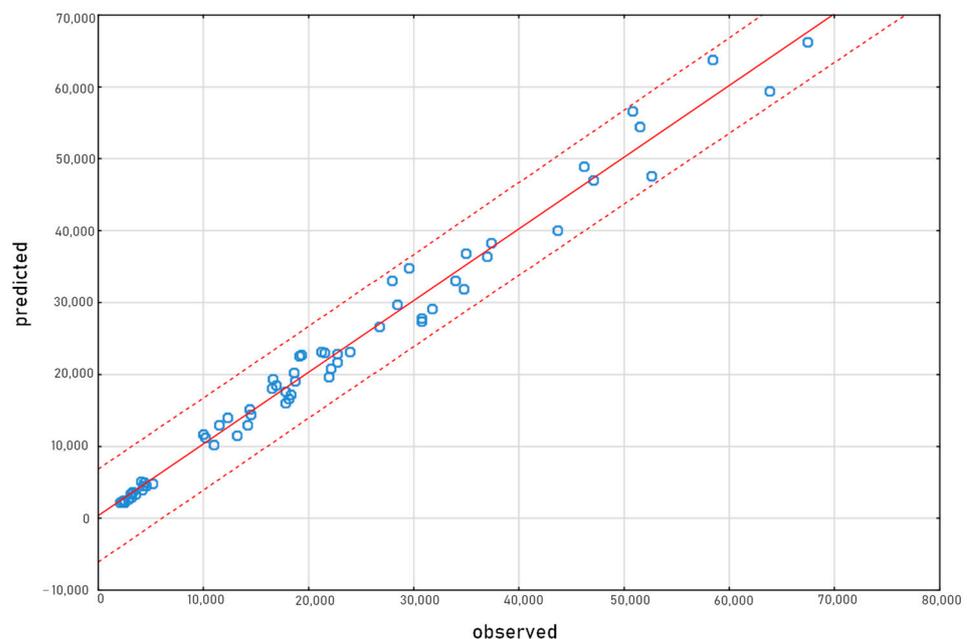


Figure 5. A scatter plot of model predictions against observed order quantities. The blue dots indicates the actual values, while the red line indicates the predicted model, while the dotted red lines indicate the confidence interval.

Figure 5 shows a graphical representation where, on the Y axis, we have the predictions made by a model and, on the X axis, we have the actual observed quantities of orders.

Lines representing confidence intervals in a scatter plot indicate the range within which the true values of the data points are likely to fall with a 90% level of confidence. In other words, they provide an estimation of the uncertainty associated with the predicted values. Typically, the narrower the confidence interval, the more precise the predictions are considered to be.

3.3. Classification of Analyzer Overload—Decision Trees and Logistic Regression

The objective was to develop models that characterize the utilization patterns of analyzers in different time intervals and create efficient models based on them that describe the temporal variability. This enables the representation and prediction of their successive values in the future. The developed mechanisms optimize the workload of analyzers. The objective of the task was to eliminate periods of excessive workload for analysts, which cause bottlenecks, accelerated wear, and delays in issuing test results. Additionally, it is essential to reduce the time during which a sample awaits the completion of the requested analysis. The workload for analysts should be evenly distributed, ensuring scheduled downtime for maintenance and other service activities related to the equipment.

The initial phase of data processing in the acquisition, parsing, and analysis process involved creating techniques to identify workload limits for analyzers, followed by optimization. These mechanisms were then tested and optimized. Based on the results, fundamental structures and interfaces were developed to signal alarm states for analyzer workload.

An analysis was conducted on the cyclic nature of workload curves for analyzers in different cycles and various devices (Figure 6). The factors and periodic indicators for each curve were evaluated to facilitate the selection of periods for training models when approximating workload curve models for analyzers.

Each of the graphs in Figure 6 present the daily cycles of variability in analyzer load for different parameters. Each of the three graphs display data from multiple daily cycles. The essence of approximating the daily curve is to find statistical characteristics that best capture the cyclical variability in load and allow for the most efficient prediction of device overload moments, as this is when process bottlenecks are expected. Alerting such a situation a few minutes in advance, before overflow occurs, allows for the relocation of some samples to another analyzer and avoids queues.

The use of approximation models facilitated a statistical study to detect the maximum performance point in the analyzer model. This analysis enables the identification of the alerting point before the peak performance point in the prepared program. The aim was to find characteristics that accurately represent the variability of the load while also presenting it in a way that ensured it was smooth and resistant to momentary disturbances and deviations. Regression tools, such as linear regression, exponential regression, polynomial regression, spline regression, and DWLS (distance-weighted least squares smoothing) [17], were used to analyze the variability of individual curves. The LOWESS (locally weighted scatterplot smoothing) method produced the most accurate curve fitting results. This method determines the individual points of the curve using polynomial regression models, resulting in a well-fitted approximation of the entire pattern in strongly nonlinear and irregular models. This captures the specific nature of the time-indexed load dependency. The LOWESS method involves fitting a regression curve to a subset of the data by selecting a window for each data point that includes only the nearest neighboring points. The window size is a hyperparameter of the model. Within this window, regression is performed using a low-degree polynomial to fit the local curve to the data. Weighted least squares are used to assign greater weight to nearby points and less weight to distant points, allowing for the consideration of local trends. The key parameter of the LOWESS method is the width of the window, which determines the number of points included in the local curve fitting. Narrower windows result in more detailed smoothing, while wider windows lead to more general trends. The LOWESS algorithm repeats this process for all data points, resulting in a smoothed curve that reflects local trends in the data. The LOWESS method is flexible and

can handle nonlinear dependencies between variables. However, the algorithm may be susceptible to the influence of outliers or be unstable in the case of low-density data [18].

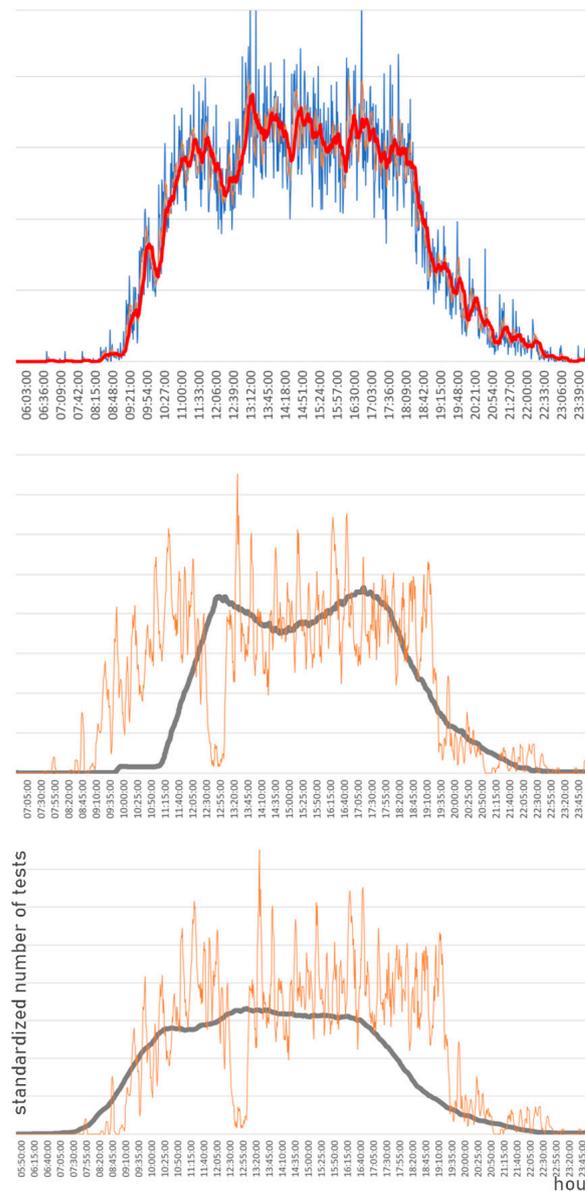


Figure 6. Approximation of models for workload curves for analyzers in different cycles for various analyzers using various statistical characteristics. In the first graph we see several waveforms (daily series) plotted on one scale (blue + orange), on the next two (orange) exemplary daily waveforms are separated.

The median was used as the most representative indicator. A rolling median was calculated based on historical data from the past 10 runs, followed by another calculation based on the median historical run with a window of 5. A regression coefficient was then computed to allow for slope assessment, assuming a zero intercept (intercept = 0) to maintain the intersection point on the same level for each line in the resulting dataset. The training data were labeled using the ‘signal’ parameter to indicate a binary label for when an alert should be triggered. This signifies the saturation of the analyzer and, specifically, an assessment of saturation risk, requiring the initiation of procedures related to redirecting further investigations to another device.

Logistic regression is a popular statistical model used for analyzing categorical data, where the dependent variable takes two possible binary values. Logistic regression is a classification technique that predicts the probability of something belonging to one of two classes based on the values of independent variables, such as characteristics in the form of a rolling median and time. The main idea behind logistic regression is to transform a linear regression model into the logarithm of odds space, which allows for the modeling of the probability of something belonging to a specific class. Unlike linear regression, which predicts continuous values, logistic regression estimates the probability of something belonging to one of two classes. The basic form of logistic regression is a binary model where the dependent variable takes values 0 or 1. This model is based on the logistic function, also known as the sigmoid function, which transforms the results of linear regression into the $[0, 1]$ range [19].

The logistic regression model achieved a satisfactory 96% correct classification rate. Other devices achieved even higher accuracy, with the largest analyzers reaching up to 99.5%. However, some analyzers had an error rate as high as 15.1%, indicating weaknesses in the model that require the careful monitoring of curve variability in the lower range. The error occurred due to temporary, minor intensities in the daily cycle. To address this issue, it is necessary to narrow the variability range in daily cycles, particularly during hours when loading the analyzer to the narrow throat level is not possible. However, calibrating each analyzer model separately is not optimal or feasible when the set of analyzers subjected to such prediction is not closed and constant.

To enhance result accuracy, we decided to use a different classifier, specifically a decision tree, along with the slope metric of the curve represented by calculated regression coefficients and a moving median for load profiles. CART (Classification and Regression Trees) models are a widely used method for classification and regression in data analysis. Decision trees are diagrams that show hierarchical decision structures. They allow decisions to be made based on a set of conditions or independent variables. CART constructs a decision tree by dividing the dataset into subsets based on the values of independent variables to minimize heterogeneity (impurity) within each subset [20].

The aim of the optimization was to decrease the maximum load analyzer time by 5% by redirecting a portion of the samples to other devices when maximum efficiency was achieved, with a minimum accuracy of 80%. To prevent systematic errors associated with seasonal loads, a sample of 30 daily profiles was randomly selected from five random analyzers in the most burdened group. The data for one day of work of the five analyzers were collected over a period of at least three months, considering the years 2020 and 2021. As a control group, 30 random daily profiles were selected from the two previous years during the same period of the year. This corresponds to three months and five random analyzers.

Table 3 presents the results of the conducted tests. The value referred to as the “average daily load” essentially represents the percentage of time when the device operates at maximum capacity. This is evident from the characteristic plateau observed in the graphs in Figure 6, indicating that the device has a backlog of orders and the samples are “waiting” for analysis. In this context, the standard deviation shows by how many percentage points these average values typically differ across different daily cycles. The reduction in the load, induced by transferring samples to other devices after the forecasted overload alert is observed, is measured by the “average daily load (Δ transfer)” variable. In this case, the standard deviation does not significantly differ from the results without transfer.

The accuracy of alert predictions, measured as the percentage agreement with load predictions, was calculated for ten of the most heavily loaded analyzers, as well as for 30 daily runs randomly selected for the years 2021 and 2022. The accuracy level exceeded 80% in each analyzer’s case, with an overall accuracy of 81.72%. The tests showed that each analyzer reduced working time by more than 10%, with an average reduction of 11.98% under maximum load.

Table 3. The results of the testing procedure.

| Analyzer ID | % of Load Prediction Accuracy | Average Daily Load | Standard Deviation of Daily Load | Average Daily Load (Δ transfer) | Standard Deviation of Daily Load (Δ transfer) | % Reduction in Load Time (Δ transfer) |
|-------------|-------------------------------|--------------------|----------------------------------|---|---|---|
| 1442906466 | 81.74% | 48.04% | 0.10 | 41.09% | 0.08 | 14.22% |
| 1460164480 | 83.67% | 46.33% | 0.12 | 40.87% | 0.11 | 11.65% |
| 780608985 | 82.84% | 36.23% | 0.11 | 32.00% | 0.10 | 11.62% |
| 818173510 | 80.82% | 61.86% | 0.15 | 53.83% | 0.13 | 13.01% |
| 818173511 | 82.02% | 65.25% | 0.13 | 57.44% | 0.11 | 11.99% |
| 923237128 | 82.36% | 47.89% | 0.10 | 41.71% | 0.08 | 12.90% |
| 923237129 | 81.48% | 49.96% | 0.11 | 40.87% | 0.11 | 11.59% |
| 2535034200 | 80.02% | 40.25% | 0.11 | 32.00% | 0.1 | 10.26% |
| 171978668 | 80.52% | 63.74% | 0.1 | 53.83% | 0.13 | 11.04% |
| 797783690 | 81.69% | 67.22% | 0.12 | 57.44% | 0.11 | 11.54% |
| Total | 81.72% | 52.68% | 0.12 | 45.11% | 0.11 | 11.98% |

3.4. Detection of the Required Calibration Moment—Neural Networks

The initial step is to identify weaknesses in the control processes, such as quality control parameters that do not accurately reflect the true variability of the measurements of the analyzers due to small initial study samples and the excessive precision of measurements on control materials. Further efforts should focus on identifying mechanisms to achieve quality control by determining appropriate methods for analyzing data from real patient measurements [21]. Quality control (QC) based on measurements of proprietary control materials covers only the analytical stage of the result generation process. Patient-based quality control techniques have been described for over fifty years and have been widely used in hematology for forty years [22]. However, due to practical issues, they are not widely applied in clinical chemistry laboratories. Nevertheless, recently, due to the availability of intermediary software and a greater appreciation of the benefits of these processes, there has been an interest in exploring their use as quality control tools. One method of such analysis is the assessment of “averages of normals” (AONs) proposed by Cembrowski.

The purpose of implementing these assumptions was to create models for estimating the compatibility of statistical methods with the actual results obtained on analyzers during quality control studies and after device recalibration. One of the elements involves identifying the parameters of the AoN method [23].

- Exclusion from the set of measurement values that fall outside the reference ranges [24].
- Determining the number of results necessary for calculating averages (e.g., using the Cembrowski procedure).
- Establishing control ranges.
- Developing a model to assess the impact of recalibrating analyzers on the statistical characteristics of the results obtained on the devices [25].

MA QC, also known as Patient-Based Real-Time Quality Control (PBRTQC), is a mathematical procedure that averages patient test results in real time and uses the obtained mean values for quality control purposes [26]. Patient-based QC generally uses the mean, but other algorithms, including the median, exponentially weighted moving average, and others, have also been developed and evaluated. The effectiveness of methods based on patient results depends heavily on the selected cutoff levels. Reference ranges have been studied for decades, yet there is still no effective and universal method for determining cutoff points, as noted by Cembrowski [27]. It is important to use precise technical terms and avoid figurative language when discussing scientific concepts. The degree of interpersonal variability in a measured analyte, or the variance at the interindividual level, plays a significant role.

In laboratory practice, quality control (QC) tests are routinely conducted both daily in the morning and when there is a change in Lot or Batch, referring to the reagent used for analyses on the device. Recalibration is performed when QC checks reveal any discrepancies, as well as according to a predetermined schedule. The aim of the study in the reported task was to identify which quality controls, especially recalibrations, were unnecessary. The need for corrective action resulting from inaccurate quality control outcomes can be verified or refuted based on patient test results (see Figure 7).

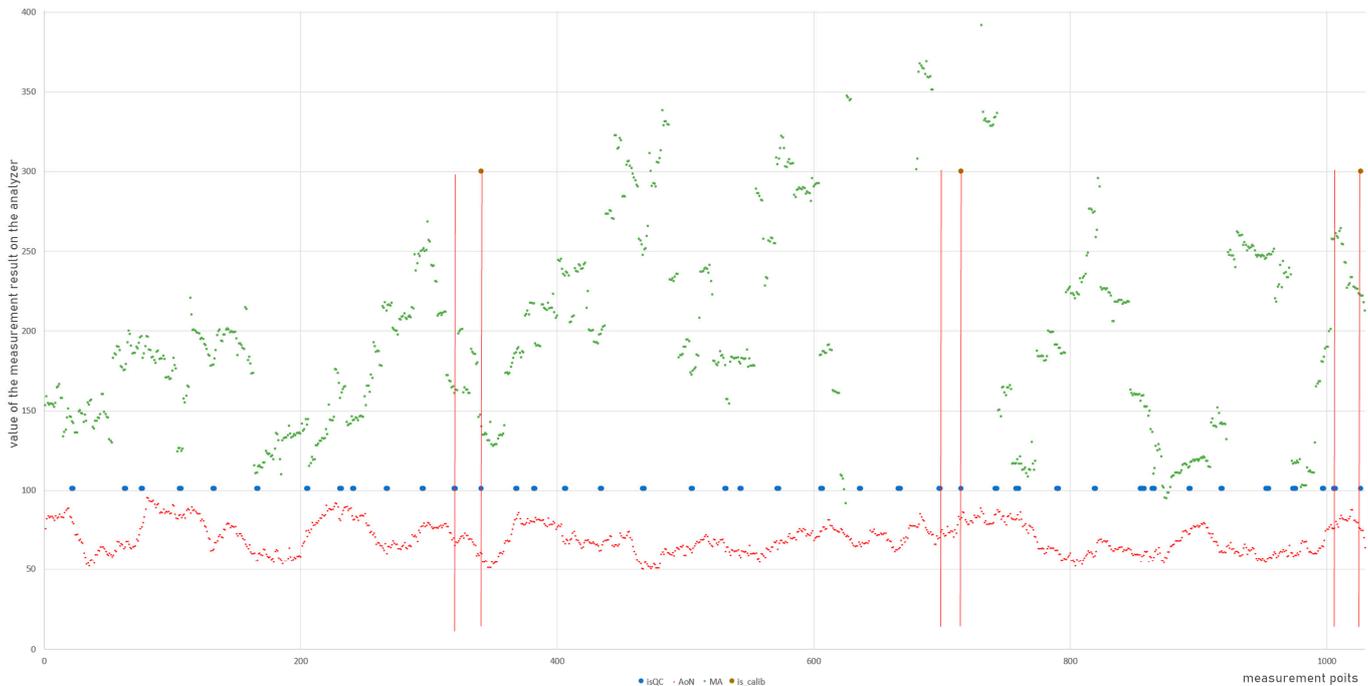


Figure 7. Quality control and calibration points in relation to AoN and MA. The pairs of vertical lines: the first indicating quality control and the second (marked with the dot on the top of line) marking the calibration of the device as a result. The remaining blue dots at the 100 level represent quality control marks that did not result in the need for calibration.

Figure 7 shows a variability plot indicating that the MA results exhibit increasing variability over time (rather than AoN), leading to a quality control, followed by calibration. They are marked by pairs of vertical lines, with the first indicating quality control and the second (marked with the dot on the top of line) marking the calibration of the device as a result. The remaining blue dots at the 100 level represent quality control marks that did not result in the need for calibration; hence, from the perspective of data quality, they were not essential. The aim is to capture the change that necessitates calibration, thus demonstrating the absence of the need for quality control in the remaining points.

Figure 7 presents the actual values of the tested material from patients in the form of data points. The diversity of results is natural, reflecting the variety of disorders and individual characteristics. However, the assumptions of methods based on patient results suggest that despite this diversity, there should be constant characteristics describing the distribution of this diversity. On the graph, we can observe quality control points, marked as blue dots, and device recalibration, marked as red vertical lines.

Statistical algorithms used for quality control on patient data include the following:

1. AoN (Analysis of Numbers).
2. Moving average of test results in a block.
3. Moving average of natural logarithms of test results in a block.
4. Moving average of square roots of test results in a block.
5. Moving median of test results in a block.

6. Moving median of natural logarithms of test results in a block.
7. Moving median of square roots of test results in a block.

Figure 8 presents the actual values of the examined material from patients as points, while various statistical characteristics are represented by dashed lines. The purpose of this chart is to magnify a short segment of data to observe the change in the curves resulting from device calibration. This change suggests that calibration affects the characteristics of the distribution of test result variability. To investigate how calibration affects parameters, individual points were labeled with 'DIFF' labels. These labels take the value of 1 when a difference is observed (Figure 8) and 0 when calibration does not alter the indicators.

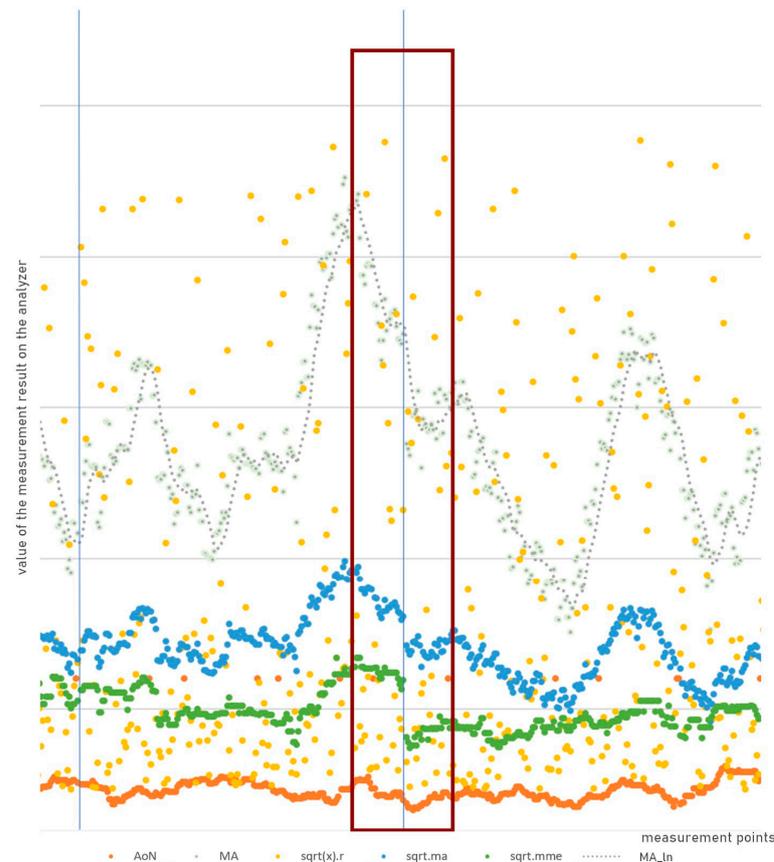


Figure 8. The visible changes in indicators after calibration: actual values of the examined material from patients are presented as points, while various statistical characteristics are represented by dashed lines. The brown box marks the area before and after calibration (indicated with blue vertical line) where the change in indicator values is best visible.

This labeling allows us to distinguish between the observation blocks from before and after calibration (Figure 9). As observed, the results after calibration (first two box plots) do not differ significantly—their boxes overlap. However, the values in the subsequent two plots show significant variation. Therefore, the conclusion is that observations in blocks preceding the calibration, marked as DIFF -1 (i.e., necessary), are significantly different from those preceding the calibration, marked as DIFF 0 (i.e., not introducing a significant change).

The analysis visualized in Figure 9 aimed to examine whether the selected characteristic, depicted here as the Moving Average of Square Roots of test results in a block, exhibits significant differences between values before and after calibration. Calibrations labeled as DIFF = 1, meaning those deemed necessary from a quality control standpoint, show a significant difference in the presented characteristic.

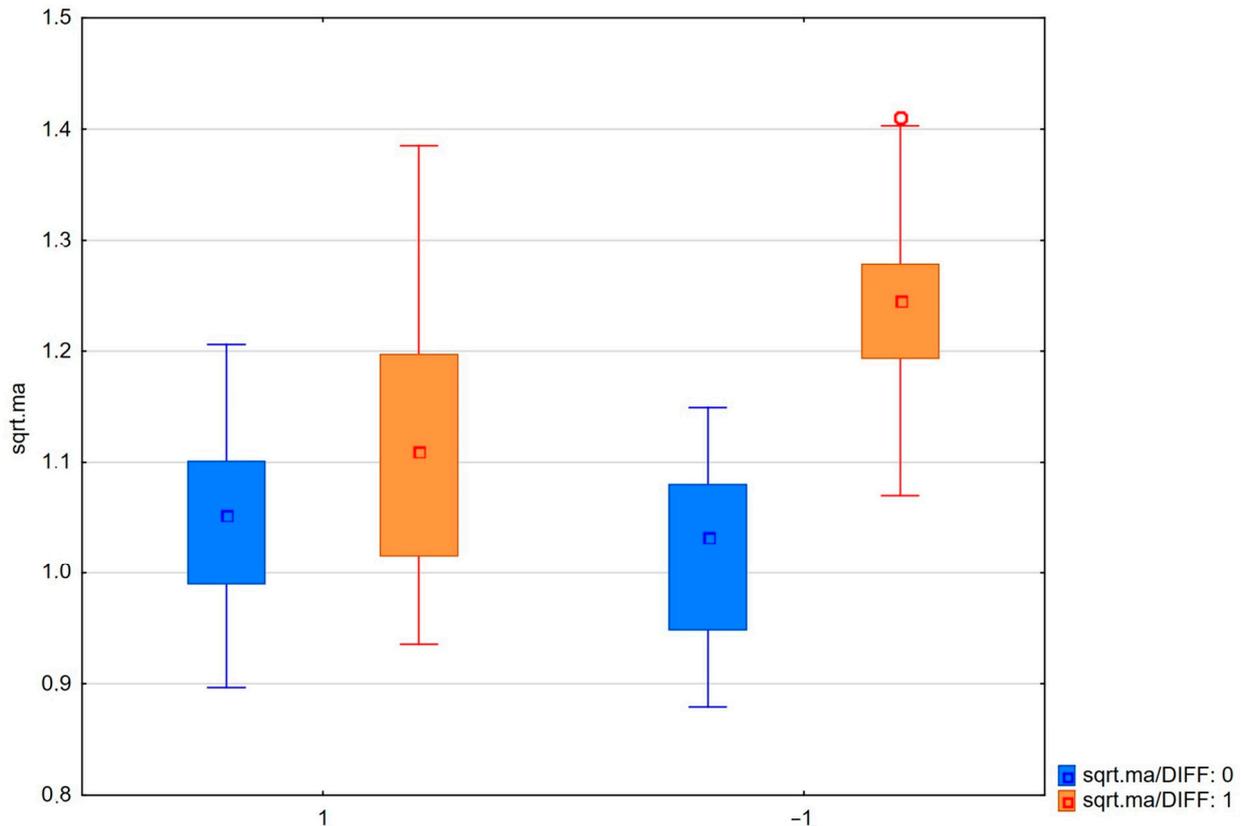


Figure 9. Box-and-whisker plots (boxplots) for sqrt (MA) for groups DIFF 0 and DIFF 1 in the context of the periods before and after calibration.

The goal of the further models was to capture the dependencies of indicator impact on the calibration designation as necessary (DIFF = 1) or unnecessary (DIFF = 0), specifically identifying quality control points that could potentially be omitted since they are not essential. To achieve this, three models were developed: the first utilized decision trees, and the other two employed neural network models.

The developed decision tree model indicates that, based on the two most important indicators, AoN and MA, an effective classifier can be created to distinguish between necessary and unnecessary calibrations without introducing significant changes. However, it was also demonstrated that the model is not entirely accurate. While many leaves exhibit high accuracy at 87% and 95%, there is a node that serves as a partition with high variability, associated with an error rate of 43% (for high values of moving averages and AoN when the device approaches a state requiring calibration). Hence, the employment of alternative techniques was deemed a necessity.

The next algorithm used in the classification of the control points was a neural network model, which yielded significantly better fitting compared to the decision trees. We utilized the Automatic Neural Network (ANN from Statistica 13.1 Statsoft Poland) algorithm for architecture searching and exploring the spaces of possible architectures, including varying the number of layers and neurons in the hidden layers, as well as MLP and RBF architectures, different activation functions, determining the optimal architecture to be MLP 9-13-2 (Statistica 13.1, Statsoft, Kraków, Poland). The MLP 9-13-2 configuration specifies the architecture of the network, indicating that it consists of 9 input neurons (determined by input variables), a hidden layer with 13 neurons, and 2 output neurons. Also, the notation BFGS 92 refers to the Broyden–Fletcher–Goldfarb–Shanno training algorithm with 92 cycles of iteration. This optimization algorithm is used to minimize the objective function during the training process. The BFGS algorithm is a quasi-Newton method that approximates the second derivative (Hessian matrix) of the objective function and updates the parameters

iteratively to find the minimum of the function. SOS (“Sum of Squares”) and entropy error functions are functions used in the context of neural networks for error calculation during the training process. Since the network architecture was generated automatically and the best configurations were tested, we can observe that neither error metrics nor the types of activation functions had a significant impact on the network’s results. As evident from these comparisons, activation functions perform similarly regardless of whether they are hyperbolic tangents (tanh), exponential functions, or softmax functions. Numerous models were created, and the best exemplary architectures with good results are presented in Table 4. The presented network architectures differ in accuracy on the validation set, with noticeable differences. Interestingly, the networks labeled ID 2 and ID 5 exhibited higher effectiveness despite weaker performance on the test set, suggesting overfitting in network ID 4.

Table 4. The best network architectures.

| Network ID | Network Name | Training Quality | Testing Quality | Validation Quality | Learning Algorithm | Error Function | Hidden Activation | Output Activation |
|------------|--------------|------------------|-----------------|--------------------|--------------------|----------------|-------------------|-------------------|
| 2 | MLP 9-13-2 | 99.13% | 94.52% | 98.63% | BFGS 92 | SOS | Tanh | Exponential |
| 4 | MLP 9-12-2 | 100.00% | 100.00% | 97.26% | BFGS 71 | Entropy | Tanh | Softmax |
| 5 | MLP 9-13-2 | 99.13% | 93.15% | 98.63% | BFGS 25 | Entropy | Exponential | Softmax |

Our observations reveal that a smaller number of input neurons, representing only two variables—AoN and MA—resulted in a decrease in accuracy. Hence, it can be inferred that utilizing additional indicators allows for better prediction. A sensitivity analysis of the network enables the assessment of the impact of individual indicators on the network output. The results can be compared by carrying out an evaluation using CART. As depicted in Table 5, neural networks exhibit a preference for MA over AoN, although $\ln(\text{MMe})$ also proves to be a significant factor. A detailed sensitivity analysis can be seen in Table 5. It includes assessments of the impact of individual factors (significance) on the results obtained by the networks. The analysis showed that $\ln(\text{MMe})$ —moving median natural logarithms—takes, on average, third place in the ranking of the influence of individual parameters.

Table 5. Sensitivity analysis of the best network architectures.

| Network | MA | $\ln(\text{MMe})$ | Sqrt(MA) | AoN | $\ln(\text{MA})$ | sqrt(MMe) | MM.median | $\ln(x).r$ | sqrt(x).r |
|--------------|----------|-------------------|----------|-------|------------------|-----------|-----------|------------|-----------|
| 2.MLP 9-13-2 | 26,960.4 | 255 | 289.3 | 6 | 13 | 56.5 | 42.1 | 1.3 | 1.1 |
| 4.MLP 9-12-2 | 443.4 | 167.8 | 41.6 | 120.9 | 92.4 | 33.7 | 19.1 | 3.8 | 2.2 |
| 5.MLP 9-13-2 | 4.2 | 14.3 | 1.7 | 2.7 | 5.9 | 4.1 | 1.2 | 1 | 1.1 |
| Average | 9136 | 145.7 | 110.9 | 43.2 | 37.1 | 31.4 | 20.8 | 2 | 1.5 |

Table 5 indicates yet another aspect of comparing the presented architectures. Each of these selected networks has a different proportion of influence on individual characteristics. Network ID 2 strongly prefers the moving average, with its importance being two orders of magnitude greater than the next characteristic. On the other hand, the last network, ID 5, prioritizes the natural logarithm of the moving median, followed by the natural logarithm of the moving average, but the differences in importance are not as pronounced. Only network ID 4 assigns significant weight to AoN.

The error that was particularly crucial in this situation, namely the False Negative, indicates that the model predicted that calibration was unnecessary when, in fact, it was essential. This error stands at 1.5% in the best model, while the overall accuracy of the model is 98.8%.

4. Discussion

The results of the presented research work share a common goal and foundation. As a system of interconnected models, they enable the optimization of diagnostic laboratory net-

work operations. They implement the functional assumptions of multiple analytical models, collectively referred to as CM4QC—Computational Model for Quality Control. However, a significant distinction among the described models lies in their shared characteristic—instead of the need to power each functionality with a separate database obtained by specialists from different departments, we introduce a common data source. Additionally, these data comprise the fundamental raw material for laboratories—information derived from real patient test results. In this way, machine learning models facilitate the rapid and cost-effective improvement of quality management processes using readily available resources. The concept of the system is summarized in Figure 10.

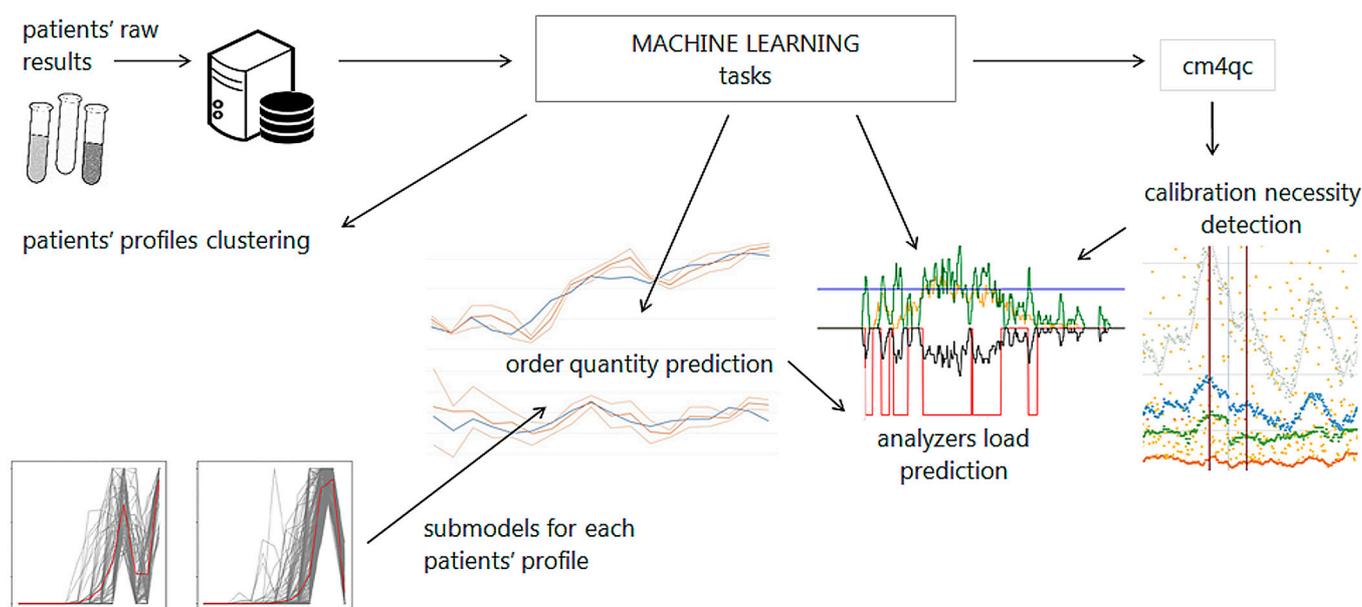


Figure 10. Examples of machine learning algorithms results from individual tasks in the research.

As sub-models of the system, we identify the following:

- SOM networks (patient segmentation), which enable the creation of predictive models for the number of orders for each profile separately.
- Exponential smoothing was employed for prediction.
- Simple multilayer perceptrons (MLPs) were utilized to predict the need for analyzer calibration.
- Logistic regression was applied to predict the moment of device filling.
- Decision trees were used for classification.

As noted, a variety of straightforward machine learning techniques rooted in data from research results were employed. The achieved results, characterized by high accuracy and reliability, enable effective decision making. Such an information flow structure yields simple alerts for matters such as device fill levels, calibration needs, or order volume for reagent inventory management. However, the decisions obtained enable the more effective and faster management of individual laboratory centers. The implemented machine learning models have increased laboratory efficiency.

The demonstration included the accuracy of alert predictions, measured by the percentage agreement with load predictions, for ten analyzers experiencing the highest workloads. This evaluation covered 30 randomly selected daily runs for the years 2021 and 2022. For each analyzer, accuracy surpassed 80%, achieving an overall accuracy of 81.72%. Notably, during the tests, each analyzer exhibited a time reduction exceeding 10%. This resulted in an average decrease in working time under maximum load by 11.98%. These are tangible time savings, and most importantly, decisions are made based on ‘live’ data already possessed by the company.

Practical signs of laboratory performance improvement could be observed after the introduction and implementation of the proposed enhancements into production practice. However, at this point, we can only discuss research results, even when referring to the validation of results on a large number of cases. Nonetheless, we can gather and summarize the achieved results, which are highly promising:

Order Size Prediction Model: It turns out that according to the ultimately adopted solution, obtaining an order plan for 10 analytes, or even 11 analytes (which could potentially be more), takes a fraction of a second. Despite the fact that preparing the models took many weeks and included selecting the methodology, developing profiling, clustering, selecting the forecasting algorithms and aggregation methods, determining the hyperparameter values of the models, and, finally, developing the equations of the models themselves, these ready-made prediction models can be calculated as a set of equations with a sliding time window for prepared time series, and obtaining results takes less than a second—the same as updating a spreadsheet. For these finished models, computation time is an insignificant boundary condition. Obtaining the order plan for analytes can be achieved extremely quickly with such a tool. The consumption balance of reagents for testing can be calculated as a direct ratio, where one unit of reagent is consumed per test, so the order plan is simply a comparison of the forecasted monthly order numbers for each analyte. Some reagents are consumed in quantities above 30,000 units per month, while others are consumed in quantities below 2000 per month. Therefore, calculating the MAPE for each separately and then averaging it, the average error will be only 2.16%, indicating the precision of the models. These forecasts (for each analyte for 6 months), treated separately, constitute the inventory and order plan. We can indicate the precision and effectiveness of the models, but the realization of savings generated by reducing the level of reagent inventory based on these forecasts could only be calculated after implementing these methods into the company's logistics procedures, but currently, we do not have data on this.

Calibration Necessity Analysis Model: The error that was particularly important in this situation, namely False Negative, indicates that the model predicts that calibration is unnecessary when it is actually required. This error in the best model is 1.5%, while the total accuracy of the model is 98.6%. Even estimating with a large margin of error (much higher than that quoted for network models) at the level of 10%, we can predict 90% out of 11% of the average number of tests performed, which constitutes quality control, calibrations, and repeats. However, at this stage, it cannot be determined whether this knowledge can be utilized every time by foregoing unnecessary calibrations and thus saving time and materials, as these decisions also depend on legal and procedural considerations.

Analyzer Work Time Optimization Model: In the case of analyzer overload, the parameter assumed a 5% reduction in the analyzer's working time under maximum load. In the tests, each analyzer showed a reduction in time of over 10%. On average, the working time under maximum load was reduced by 11.98%. On average, this group of analyzers performs approximately 23,000 tests per day. After implementing the optimization, the number of tests performed without delays can even exceed 25,000 tests per day.

5. Conclusions and Future Research

This study proposes the development of the PBRTQC approach [17,21], which stands for Patient-Based Real-Time Quality Control, for a broader spectrum of applications than just quality control itself. Based on data from real-time patient studies, we can draw conclusions regarding quality control, as well as reagent management, analytical device overload, and patient profiles. The scientific novelty of this study lies in the fact that it presents a comprehensive approach to utilizing additional knowledge from ongoing analytical research and the very concept of the solution. Hence, there is less urgency to search for unique machine learning models, as our aim was to instill trust in results derived from relatively easy-to-interpret models that are also explainable to users whose domain is not computational models but traditional statistical quality control techniques. However, if such a solution gains full user trust, we can then explore areas for improvement in

individual modules and strive to increase precision using modern, more sophisticated machine learning models for further studies. Research also indicates that there is further potential for system development. The obtained patient profiles may, in the future, allow for individual analyses of patients' test results to personalize recommendations for further examinations. However, this is a topic for future research.

Author Contributions: Conceptualization, K.R., A.O. and P.W.; data curation, P.W., J.S. and P.S.; formal analysis, K.R. and A.O.; funding acquisition, A.K.-Ś., P.W. and J.S.; investigation, K.R., A.O. and P.W.; methodology, K.R. and A.O.; project administration, A.K.-Ś.; supervision, P.W. and J.S.; validation, P.S.; writing—original draft, K.R. and A.O.; writing—review and editing, K.R. and A.O. All authors have read and agreed to the published version of the manuscript.

Funding: The research was co-financed by the European Regional Development Fund under the Intelligent Development Program 2014–2020, project “Automatic prediction system for laboratory research and optimization of quality control and workload of analyzers based on machine learning”, funded as part of a competition organized by the National Centre for Research and Development under grant POIR.01.01.01-00-1259/20-01 and Diagnostyka S.A.

Data Availability Statement: Data is contained within the article.

Conflicts of Interest: Authors J.S., P.S., P.W. and A.K.-Ś. were employed by the company Diagnostyka S.A. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

References

1. Badrick, T.; Bietenbeck, A.; Cervinski, M.A.; Katayev, A.; van Rossum, H.H.; Loh, T.P. Patient-Based Real-Time Quality Control: Review and Recommendations. *Clin. Chem.* **2019**, *6*, 962–971. [[CrossRef](#)] [[PubMed](#)]
2. Zhou, R.; Wang, W.; Padoan, A.; Wang, Z.; Feng, X.; Han, Z.; Chen, C.; Liang, Y.; Wang, T.; Cui, W.; et al. Traceable machine learning real-time quality control based on patient data. *Clin. Chem. Lab. Med. (CCLM)* **2022**, *60*, 1998–2004. [[CrossRef](#)] [[PubMed](#)]
3. Lu, Y.; Yang, F.; Wen, D.; Shi, K.; Gu, Z.; Lu, Q.; Wang, X.; Dong, D. Assessment of patient based real-time quality control on comparative assays for common clinical analytes. *J. Clin. Lab. Anal.* **2022**, *36*, e24651. [[CrossRef](#)] [[PubMed](#)]
4. Liang, Y.; Padoan, A.; Wang, Z.; Chen, C.; Wang, Q.; Plebani, M.; Zhou, R. Machine learning-based nonlinear regression-adjusted real-time quality control modeling: A multi-center study. *Clin. Chem. Lab. Med. (CCLM)* **2024**, *62*, 635–645. [[CrossRef](#)]
5. Liang, Y.; Wang, Z.; Huang, D.; Wang, W.; Feng, X.; Han, Z.; Song, B.; Wang, Q.; Zhou, R. A study on quality control using delta data with machine learning technique. *Heliyon* **2022**, *8*, e09935. [[CrossRef](#)] [[PubMed](#)]
6. Li, K.; Sward, K.; Deng, H.; Morrison, J.; Habre, R.; Franklin, M.; Chiang, Y.-Y.; Ambite, J.L.; Wilson, J.P.; Eckel, S.P. Using dynamic time warping self-organizing maps to characterize diurnal patterns in environmental exposures. *Sci. Rep.* **2012**, *11*, 24052. [[CrossRef](#)]
7. Javed, A. SOMTimeS: Self Organizing Maps for Time Series Clustering and its Application to Serious Illness Conversations. *arXiv* **2021**, arXiv:2108.11523. [[CrossRef](#)]
8. Fix, J.; Frezza-Buet, H. Look and Feel What and How Recurrent Self-Organizing Maps Learn. Advances in Self-Organizing Maps, Learning Vector Quantization, Clustering and Data Visualization. *Adv. Intell. Syst. Comput.* **2020**, *976*, 3–12. [[CrossRef](#)]
9. Pei, J.; Han, J.; Lakshmanan, L.V.S. Mining Sequential Patterns by Pattern-Growth: The PrefixSpan Approach. *IEEE Trans. Knowl. Data Eng.* **2004**, *16*, 1424–1440.
10. Lee, B.; Puri, N.; Patel, S. 1427: Net fluid balance effect on mortality in septic Icu patients with diabetes. *Crit. Care Med.* **2022**, *50*, 716. [[CrossRef](#)]
11. Lipton, Z.C.; Berkowitz, J.; Elkan, C. A critical review of recurrent neural networks for sequence learning. *arXiv* **2015**, arXiv:1506.00019.
12. Graves, A.; Mohamed, A.R.; Hinton, G. Speech recognition with deep recurrent neural networks. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Vancouver, BC, Canada, 16–31 May 2013; pp. 6645–6649.
13. Nowak, J.; Korytkowski, M.; Scherer, R. Discovering Sequential Patterns by Neural Networks. In Proceedings of the International Joint Conference on Neural Networks (IJCNN), Glasgow, UK, 19–24 July 2020; pp. 1–6. [[CrossRef](#)]
14. Weiping, T. RNN-based signal classification for hybrid audio data compression. *Computing* **2019**, *102*, 813–827.
15. Mills, T.C. ‘Classical’ Techniques of Modelling Trends and Cycles. In *Modelling Trends and Cycles in Economic Time Series*; Palgrave Macmillan: Cham, Switzerland, 2021. [[CrossRef](#)]
16. Hyndman, R.J.; Athanasopoulos, G. *Forecasting: Principles and Practice*; Textbook OTexts: Melbourne, Australia, 2018; Available online: <https://otexts.org/fpp2/> (accessed on 1 February 2024).

17. Yang, Y.; Wu, D.; Zeng, L.; Li, Z. Weighted least square filter via deep unsupervised learning. *Multimed. Tools Appl.* **2023**, *83*, 262030100. [[CrossRef](#)]
18. Nordhausen, K.; Taskinen, S. Locally Weighted Scatterplot Smoother. In *Encyclopedia of Mathematical Geosciences*; Daya Sagar, B.S., Cheng, Q., McKinley, J., Agterberg, F., Eds.; Springer: Cham, Switzerland, 2021. [[CrossRef](#)]
19. Lever, J.; Krzywinski, M.; Altman, N. Logistic regression. *Nat. Methods* **2016**, *13*, 541–542. [[CrossRef](#)]
20. Opalinski, A.; Regulski, K.; Mrzyglod, B.; Glowacki, M.; Kania, A.; Nastalek, P.; Celejewska-Wójcik, N.; Bochenek, G.; Sladek, K. Medical Data Exploration Based on the Heterogeneous Data Sources Aggregation System. In Proceedings of the 2019 Federated Conference on Computer Science and Information Systems ACSIS, Leipzig, Germany, 1–4 September 2019; Ganzha, M., Maciaszek, L., Paprzycki, M., Eds.; Volume 18, pp. 591–597. [[CrossRef](#)]
21. Cembrowski, G.S.; Cervinski, M.A. Demystifying Reference Sample Quality Control. *Clin. Chem.* **2016**, *62*, 907–909. [[CrossRef](#)] [[PubMed](#)]
22. Badrick, T.; Cervinski, M.; Lohc, T.P. A primer on patient-based quality control techniques. *Clin. Biochem.* **2019**, *64*, 1–5. [[CrossRef](#)] [[PubMed](#)]
23. Cembrowski, G.S.; Chandler, E.P.; Westgard, J.O. Assessment of “Average of Normals” quality control procedures and guidelines for implementation. *Am. J. Clin. Pathol.* **1984**, *81*, 492–499. [[CrossRef](#)]
24. Janecki, J. Application of statistical features of the Gaussian distribution hidden in sets of unselected medical laboratory results. *Biocybern. Biomed. Eng.* **2008**, *28*, 71–81.
25. Fleming, J.K.; Katayev, A. Changing the paradigm of laboratory quality control through implementation of real-time test results monitoring: For patients by patients. *Clin. Biochem.* **2015**, *48*, 508–513. [[CrossRef](#)]
26. van Anel, E.; Henricks, L.M.; Giliams, A.P.M.; Noordervliet, R.M.; Mensink, W.J.; Filippo, D.; van Rossum, H.H.; Cobbaert, C.M.; Gillis, J.M.E.P.; Schenk, P.W.; et al. Moving average quality control of routine chemistry and hematology parameters—A toolbox for implementation. *Clin. Chem. Lab. Med.* **2022**, *60*, 1719–1728. [[CrossRef](#)]
27. Cembrowski, G.S.; Xu, Q.; Cervinski, M.A. Average of Patient Deltas: Patient-Based Quality Control Utilizing the Mean Within-Patient Analyte Variation. *Clin. Chem.* **2021**, *67*, 1019–1029. [[CrossRef](#)] [[PubMed](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.