Hybrid Clinical Decision Support for Cardiology: Architectural Foundations for Integrations

Karina J. Shakhgeldyan[a,b][0000-0002-4539-685X], Valeriya V. Gribova[a,c][0000-0001-9393-351X], Boris I. Geltser [a][0000-0002-9250-557X], Elena A.Shalfeyeva[a,c][0000-0001-5536-2875], Bogdan V. Potapenko[*b][0009-0000-0018-4370]

[a]Far Eastern Federal University, FEFU Campus, 10 Ajax Bay, Russky Island, Vladivostok 690922, Russia; [b]Vladivostok State University 41 Gogolya st., Vladivostok 690014, Russia; [c]Institute of Automation and Control Processes, 5, Radio st., Vladivostok 690041, Russia

ABSTRACT

Cardiovascular diseases (CVD) are the leading cause of death in most countries around the world, making the accurate assessment of risks and the selection of individual preventive strategies a current focus in healthcare. In this article, the authors presented a prototype of a Clinical Decision Support System (CDSS) for predicting and preventing cardiovascular risks based on a hybrid architecture that integrates machine learning models and ontological knowledge bases. A microservice architecture based on the Cloud-Edge approach is proposed for optimizing computational resources when processing tabular data, signals, video, and images, as well as for enhancing the effectiveness of integration with various Healthcare Information Systems (HIS). The CDSS supports the formalization not only of medical history data and results of studies but also the rules for interpreting the results of predictions based on machine learning models and methods of explainable artificial intelligence (XAI). The developed CDSS includes widely used tools in clinical cardiology and cardiothoracic surgery for risk assessment, as well as proprietary machine learning models for predicting in-hospital mortality, and others. These models contribute to making informed medical decisions for the diagnosis, prevention, and treatment of CVD. The prototype was implemented at the Medical Center of the Far Eastern Federal University and integrated with the "1C" HIS. The experience of implementing the prototype demonstrated the high potential of hybrid CDSS based on microservice architecture for use in clinical practice.

Keywords: Clinical Decision Support System, Hybrid Artificial Intelligence, Microservice Architecture, Machine Learning, Knowledge Base, Explainable Artificial Intelligence, Predictive model

1. INTRODUCTION

The application of information technologies in clinical medicine includes the use of healthcare process management systems, which encompass Healthcare Information Systems (HIS) and Clinical Decision Support Systems (CDSS). They are designed to reduce the risk of medical errors, expand diagnostic capabilities, personalize the prediction of adverse events, improve management quality, and more [44]. The development of CDSS has a long history and is associated with dominant trends in artificial intelligence, architectural solutions in corporate information systems, and clinical health system practices. For a long time, CDSS was developed based on knowledge management in the format of expert intelligent systems. In recent years, CDSS is increasingly associated with systems using predictive and diagnostic models of machine learning [42].

Technologically, several CDSS architectures are distinguished: autonomous models, systems integrated into HIS, standardized, and service-oriented models [46]. Autonomous systems are calculators of individual scales, where data input

*bvpotapenko@gmail.com; phone/wechat +86-1819-404-9024

is manually performed by the doctor. In modern implementations, such calculators use a service architecture in the form of SaaS (Software as a Service), which does not solve the problem of repeated data entry and integration with the physician's workplace. HIS integrated systems allow physicians to avoid double data entry, but they require HIS developers to have broad CDSS expertise, significantly limiting the scalability of such a system [19]. The response to this challenge was the standardization of CDSS content for simplified embedding into HIS from different manufacturers [1]. In the last two decades, new CDSS architectures have been proposed. For example, the interaction between CDSS and HIS was achieved using an application programming interface (API) [40]. Furthermore, the use of a virtual medical record was proposed, solving the problem of using different medical terms for the same concept in several HIS. Finally, an approach emerged that proposed implementing CDSS based on web services [27]. Any HIS can send a standardized protocol request to CDSS, implemented as a service. Despite its many advantages, this approach requires HIS developers to implement requests to cloud-based CDSS and transfer data from Electronic Health Records (EHR) into the system, increasing the demands on HIS suppliers and causing dissatisfaction among medical organizations.

All these CDSS models implemented the storage of formalized knowledge from individual areas of clinical medicine and allowed comparing this knowledge with clinical-functional and laboratory indicators of patients for automated decision-making in clinical practice [33].

Contemporary research is characterized by the widespread use of machine learning methods for diagnosing and predicting the development of diseases and their complications [19, 21, 26]. The use of data-driven machine learning models is a natural extension of CDSS capabilities [30]. There is an increasing need for multimodal systems that integrate clinical data and results of instrumental studies (images, signals, and video), posing the challenge of combined data processing of different types and at different system nodes. For example, in diagnosing cancer, new generation machine learning predictive and diagnostic models can be used, based on the multidimensional integration of sub-models, each built on its domain data: clinical analyses, radiological and pathological studies, genome research [8]. In analyzing gastroscopic images and videos recorded during endoscopy, part of the computational load can fall on local systems and part on cloud systems, effectively combining prompt result delivery (using edge-level), measurement quality (cloud-level) [14], and potentially reducing data transmission costs to the cloud by preprocessing and compressing data at the edge level.

The goal of our study is to develop an architecture for a hybrid multimodal CDSS that integrates intelligent systems based on knowledge, predictive machine learning models, elements of prediction result interpretation, HIS, service architecture framework, cloud solutions, and services at the edge level.

2. CURRENT STATE OF AFFAIRS

Intelligent systems include formalized knowledge bases, developed according to a certain model of representation, and inference and reasoning mechanisms [16]. These systems continue to evolve, providing physicians with up-to-date information captured in clinical guidelines and diagnostic and treatment protocols [35, 36, 41]. The advantage of such CDSS is the ability to provide clear explanations derived from formalized knowledge [45]. However, this approach is limited by simplified knowledge models that may not always reflect the complexity of real clinical scenarios. Such CDSS form explanations of the decision-making process according to their simplified knowledge model [34], but advanced knowledge models corresponding to real-world representations in this field are required for intelligent physician support.

With the growth of data in clinical medicine, including genomics and medical imaging, there is an increasing need for more complex models that can integrate and analyze multimodal data [2, 28]. The development of graph representations of knowledge and reasoning based on semantic networks and knowledge graphs is widely used in intelligent systems [47]. Semantic modeling allows for the formation of clinical knowledge as interpretable clinical guidelines, an alternative to text-based guides and machine-readable specifications of recommendations [38]. Interpretable clinical guidelines, accumulated in electronic recommendation libraries (DeGeL), are terminologically compatible with electronic patient records (EPR). To assist physicians, complex systems are often created that cover diagnosis, risk assessment, treatment, and other tasks (Cardio-S, ASMO-CARDIO, Infective Endocarditis, Cardiologs, Cardiac Care Assistant, escardio.org, HEART Pathway, Cardiovascular Disease Management Tool, FFRCT HeartFlow, Cardio-ECO, Cardio-ANTIB). These approaches provide opportunities for developing more complex and scalable CDSS, adaptable to diverse clinical needs [39, 13]. Another type of CDSS are systems that do not directly use knowledge from guidelines and clinical recommendations but apply machine learning models to support decision-making. The intensive growth of these systems is associated with the development of machine learning methods and their effectiveness in medicine [43]. These systems have shown significant potential in improving the accuracy of disease diagnosis and outcome prediction [20, 24 , 3].

However, despite numerous studies on machine learning model development, only a few have been implemented in CDSS and introduced into clinical practice [16]. As the number of ML-based CDSS grows, so does the understanding of the challenges in implementing these systems; many remain insufficiently explainable to users, reducing trust and limiting clinical application [23]. However, data-based CDSS often still do not include explainable artificial intelligence mechanisms [5].

Recently, there have been projects of modern CDSS proposing the use of hybrid architecture, integrating different methods and approaches [11]. In CDSS, there is potential for combining machine learning models and diagnostic protocol knowledge bases [30]. An important advantage of the proposed CDSS model by the authors is the support for explanation tools and organizing cyclic processes of model and knowledge updates to increase trust in the system's results. Other researchers also emphasize the need for continuous knowledge base updates through data extraction and standardization in CDSS [18].

From a technological standpoint, the use of service-oriented architecture remains attractive for modern CDSS [18, 37]. The task of integrating CDSS and HIS based on the FHIR standard is one of directions that promises to simplify this process and improve their compatibility [41]. Authors of CDSS projects developed based on microservice architecture point to increased system complexity, longer transaction times, and complications in logging multiple services. The combined use of cloud and edge computing technologies for processing medical images represents a promising direction that could provide more efficient and scalable data processing directly at the physician's workplace [6].

3. CLINICAL DECISION SUPPORT SYSTEM

In this work, we propose a solution that combines approaches at three levels: conceptual, logical, and physical. In the context of this research, the term "hybrid architecture of CDSS" refers to systems in which machine learning models and intelligent knowledge bases are integrated at the conceptual level, a microservice framework system architecture and HIS at the logical level, and cloud and local data processing at the physical level. The aim of hybrid integration is to take the best from all concepts: the flexibility and predictive ability of machine learning (ML) models, and the explainability based on knowledge of the subject area, formally represented by ontologies. Microservice architecture is a variant of SOA (Service-Oriented Architecture), in which an application is implemented as a group of small, independent components (services) with specialized business logic. Each service is loosely coupled from the others, can have its independent database, can be implemented in a different programming language, and placed on a separate server [15]. Compared to SOA, microservice architecture provides greater system flexibility and allows for scaling of individual CDSS components independently of each other, enabling multiple research teams to work together, quickly implement and evaluate new ideas and techniques, and segregate data used by the microservices [48, 25].

In this study, it is assumed that each model is implemented as a separate microservice, performing a strictly limited function. The overall task of diagnosis or prediction may include several subtasks, such as computation, data processing, explanation of results, etc. Architecturally, each of these tasks should be delegated to a separate microservice. More on this will be discussed later.

This chapter focuses on the development of a hybrid CDSS architecture. The first important feature - hybridity - is achieved through the integration of intelligent knowledge bases and machine learning models. Given the inherent complexity, interpreting the results of machine learning models requires additional research with explainable artificial intelligence methods. Subsequently, the knowledge obtained in this way can be formalized in the CDSS knowledge base and applied in explaining predictive models using the intelligent knowledge base solver. Hybrid CDSS can mitigate the disadvantages and leverage the advantages of homogeneous systems: knowledge bases provide explanations and interpretations of diagnostic results, forecasts, and recommendations, while machine learning models assess risks and predict clinical events, sometimes with accuracy surpassing expert physicians.

The second key aspect of our study is the design of a microservice architecture and its combination with the Cloud-Edge approach. The use of microservices is expected to yield positive results in scenarios involving long-term application, refinement, and maintenance of a complex multi-module system, interaction of multiple scientific and development teams, and the need for integration with other systems such as HIS, laboratory information systems, wearable patient monitoring devices, etc.

In the following sections of this chapter, we will discuss the key requirements, the concept of our system, technological aspects, and propose accompanying diagrams for a more visual representation of the key features of the architecture.

The proposed CDSS architecture, combining a microservice structure and Cloud-Edge approach, could be the solution to scalability and integration challenges, as well as reducing expenses on costly communication between services.

3.1. Key requirements

The operation of a Clinical Decision Support System (CDSS) is based on requirements that reflect the key stages of the diagnostic and treatment process [12].

These stages include assisting physicians in creating EHR, preliminary disease diagnosis based on available primary data (patient complaints, results of objective examinations, possibly results of some laboratory and instrumental studies), disease diagnosis including differential diagnosis, treatment prescription (depending on the disease – medication, rehabilitation, surgery, or their combination), patient condition monitoring, assessment of the prognosis of disease development and complications, treatment, and recovery.

Considering the functional characteristics of physicians' practical activities and the analysis of literature sources [29], an ideal CDSS should support the entire spectrum of diagnostic and therapeutic tasks or be capable of expanding it through the integration of new functionalities. The system should be implemented as a single service (within a microservice architecture) with the ability to connect new services that will refine or expand the main functionality (for example, new, proprietary disease prognosis methods) and the ability to call any of the microservices (each microservice implements a specific functionality of the system). Additionally, the system should maintain operability during internet outages. It should accommodate the processing of data from wearable patient devices and handle large volumes of data from endoscopy, radiography, etc. For diagnosis and treatment planning, knowledge contained in the clinical guidelines of the Ministry of Health should be used, with the possibility of expanding it with new proprietary diagnostic and treatment methods (for the trial and clinical testing of new diagnostic tools and treatment methods). All decisions generated by the CDSS should have an explanation and/or interpretation component. The system must have the means to integrate with various medical information systems and to form/modify knowledge bases in a structure and form understandable to subject matter experts.

A knowledge base management system is used in the CDSS for supporting regulated rules and clinical protocols, and machine learning models for applying prognostic scales. Thus, the main requirement for the CDSS is a hybrid architecture that combines intelligent systems based on knowledge and machine learning models, implemented by microservices.

It is important to note that some of the formulated requirements do not directly influence the architecture of the CDSS. In the context of the possibility of expanding functionality with new microservices, this task is transferred to their developers. The CDSS itself must be capable of supporting the expansion of such functionality.

3.2. Concept

Decision support in CDSS is facilitated through artificial intelligence methods based on data and knowledge. Data-driven methods involve intelligent data analysis, where knowledge is extracted from datasets using statistical and machine learning techniques. Knowledge-based methods encompass formalized knowledge in the form of ontological knowledge bases with a graph structure, applied to medical data to generate inferences that assist physicians in diagnostic searches and treatment and prevention recommendations. Ontological graph knowledge bases also describe knowledge necessary for generating explanations and interpretations of prognosis results, diagnosis, and prescribed treatments.

The informational components of our CDSS include ontologies, knowledge, and data. Ontologies are models of knowledge and data describing their structure and the terms in which they are formed, as well as a set of rules for knowledge generation, data formation, completeness verification, and partial semantic correctness. Knowledge consists of formally represented dependencies and causal relationships between data for solving practical medicine tasks. Data is a shared set of formalized or unformalized facts of the subject area, described according to a conceptual scheme (ontology), available for processing but without interpretative capability, which is provided by formalized knowledge. Next, let's look at both types of data.

Formalized data are organized according to the supported ontology. Examples include EHR, where each attribute is a named field (number, categorical variable, string, etc.) with additional characteristics like modality, synonymy, and multilingualism. Synonymy is crucial for CDSS architecture design, solving compatibility issues with different HIS and using different terms for identical concepts. We highlighted this requirement among the key ones in the previous section of this study.

Unformalized (or weakly formalized) data include images, textual documents, charts, diagrams, audio information, etc. Examples include results of instrumental studies requiring further specialized formalization processes. In our environment (FEFU Medical Center), the most popular ones are the results of ECG, endoscopy, and radiography.

We will consider primary medical data in CDSS to be EHR, including medical histories and ambulatory cards, or real-time observed signals from wearable patient sensors. The CDSS utilizes primary data for three purposes: upon a physician's request, the CDSS processes primary data and provides information in accordance with the request (possible diagnosis and/or request for necessary data to clarify the diagnosis, prognosis of disease development or complications, prescription of additional examinations or treatments, including medication therapy, recommendations for lifestyle and dietary changes, etc.); the formation of datasets for conducting scientific research, (re)training of prognostic and diagnostic models; and real-time monitoring of the patient's condition and alerting the physician according to triggers (detection of anomalies, prediction of the onset of an adverse condition in the near future, etc.).

Primary data in the CDSS is acquired through four methods, which include integration with HIS (data from HIS is transferred to the CDSS, where it can be stored as part of selected EHR data); importing data from external sources, manually collected or previously extracted from other HIS; entering data through the CDSS user interface, allowing the input of both partial data for request implementation and complete EHRs, medical histories, and outpatient records; and through the CDSS edge-module API from wearable devices used by patients.

Data integrated into the CDSS via HIS is used both for fulfilling physician requests and for further research. Data uploaded through the second method is primarily intended for performing data analysis, training and validating models, and verifying formalized rules. It can also be available as a base for searching clinical case analogs (precedents). Data entry through the CDSS interface is intended for verifying the operation of models and algorithms in the CDSS, as well as for physicians to simulate prognostic and diagnostic decisions in case of changes in patient risk factors and in the absence of HIS. The fourth data channel is essential for real-time monitoring of patient conditions for timely prediction of health deterioration, assessing treatment effectiveness, and proactively presenting recommendations to the treating physician.

Knowledge is also divided into three conceptual categories: formalized, semi-formalized, and non-formalized.

Formalized knowledge stores rules (causal-temporal and other dependencies) of the subject area and can be applied to formalized data, to the results of data processing, or other knowledge for decision-making. Formalized knowledge describes diagnostic protocols, treatment, clinical guidelines, etc. Some types of prognostic scales can also be described with formalized knowledge, establishing a correspondence between feature values and scores correlated with the risks of adverse events. Formalized knowledge allows for the interpretation of results and the application of knowledge to data.

Semi-formalized knowledge in the CDSS represents descriptions of machine learning predictive models, which are implemented as microservices. These models, implementing diagnosis and predicting the development of diseases and their complications, are developed using machine learning methods such as linear and logistic regressions, random forest, stochastic gradient boosting, artificial neural networks, etc.

Non-formalized knowledge of the subject area (texts of protocols and clinical recommendations, articles, manuals, etc.) includes textual, graphical, audio information in file format, which can be offered to physicians as needed.

The application of knowledge to CDSS data involves two approaches: microservices and solvers. Microservices are agents that implement computations using a machine learning model or algorithm. Solvers check the fulfillment of conditions and implement actions according to the ontological knowledge base.

Besides computational microservices, the CDSS includes system-forming services that provide message routing, integration tasks, user interface generation, etc. An important task within the microservice approach is microservice orchestration and infrastructure organization, which will be discussed in the next section.

The machine learning models implemented in the CDSS are the result of data mining conducted by the project authors or other researchers, published in scientific journals. The tasks solved by these models can be grouped as follows: (1) classification is used for diagnostic tasks, including differential diagnosis, and predicting the development of diseases and complications in the near term (up to 30 days) after cardiothoracic surgery; (2) regression is employed for modeling the duration of clinical processes, such as the length of treatment, hospitalization, risk of developing complications, and for assessing the normative values of clinical-functional indicators of patients for personalized assessment of disease severity; (3) clustering and phenotyping are applied for correlating patients with groups having similar disease progression characteristics.

Data mining, which develops these models, includes methods of statistical analysis, selection and validation of predictors, training, cross-validation and testing of models, identification of risk factors, formation of their phenotypes, assessment of the relative contribution of predictors and their combinations to the realization of the endpoint, etc.

For model development, data imported into the CDSS from HIS or external datasets are used. Information about the relationship between data and the models trained on them is stored in the CDSS when describing models for further validation and retraining. As new labeled data corresponding to the CDSS prognostic models accumulates, the used models are validated, further trained, and new versions are introduced into the CDSS. This process is not only applicable to prognostic and diagnostic machine learning models but also for validating models based on formalized rules, as well as for checking the effectiveness of clinical protocols and recommendations formalized in the CDSS knowledge bases. Constant validation and updating of models and knowledge is a recurring process, initiated when new labeled datasets are accumulated.

Integration with HIS in the CDSS can be implemented in several ways. The optimal approach allows physicians to access all necessary information for decision-making from the CDSS at their workstations. However, such embedded integration may not be supported by some HIS vendors. In this case, alternative integration schemes may be considered: the physician uses the CDSS user interface and the CDSS makes requests to HIS to obtain data from EHRs , medical histories, or outpatient cards for further processing and presenting results in the CDSS user interface. If real-time patient monitoring is required and integration with HIS is not available, the CDSS can alert the physician through other channels: SMS, an emergency dedicated email channel, or via APIs of messaging systems implemented in the healthcare facility.

The explanation of predictions and decisions by the CDSS remains a separate issue.

The explainability of AI's forecasts, recommendations, and predictions in healthcare is an unsolved yet crucial problem today[32, 31, 4].

Opinions on what constitutes an "explanation" diverge in the context of healthcare, depending on the user's interests: physicians, patients, lawyers, investors, or clinic owners[6] [17].

Therefore, we choose a formal approach, which we find more universal because it is unequivocally formalized by mathematical and statistical methods, yields predictable results, and can be further translated or formatted into any other user-friendly representation.

Thus, the task of explaining the solutions proposed by the CDSS is implemented through one of the following mechanisms: (1) sequential traversal of the knowledge base tree with conditions met for the case under analysis; (2) formalized clinical interpretation in the knowledge base of forecast or diagnostic results obtained through computational agents; (3) formalized risk factors in the knowledge base that correspond to the case under analysis and serve as predictors for prognostic models; (4) formalized relative importance of predictors in the knowledge base, calculated using explainable artificial intelligence methods; formalized phenotypes of risk factors in the knowledge base corresponding to the case under analysis.

For instance, the importance of predictors for a specific case can be calculated using one of the popular methods: SHAP or LIME (these methods have become the de facto standard in the industry, so we will not dwell on them in detail), or proprietary methods developed by the author of a corresponding microservice.

Following this, the knowledge integrated into a relevant ontology (which describes the connections between the significance of predictors and clinical decisions) is applied to data, and further recommendations are suggested based on the most significant predictors in the forecast: additional examinations (such as tests, if certain parameters exceed normal values), preventive measures, recommended treatments, etc.

In this way, in the process of presenting a solution, the CDSS also accompanies it with an explanation, which focuses the physician's attention on the most significant factors and also proposes further therapeutic and diagnostic measures.

Prognostic models also include procedures for explanation and interpretation. The prescribed treatment and preventive recommendations should be accompanied by arguments and logic of inference.

Two tasks must be solved to implement these approaches in the CDSS. The first one requires to use methods of explainable artificial intelligence and interpretable machine learning for each prognostic and diagnostic model of the CDSS. The goal is to calculate threshold values of indicators with the highest predictive potential. Basing on this further steps are: (1) form phenotypes of risk factors explaining the development of an adverse event, (2) assess the contribution of each phenotype and individual risk factors to its realization, (3) explain the reasons for deviations from phenotypes, etc.

The second task is to store the explanations and interpretations in a formalized form in accordance with the template of a specialized ontology.

It should be noted that the recommendations and knowledge formalized in ontologies may not be universal in the context of using the CDSS in different administrative-territorial units (countries), where specific health regulations are implemented.

However, through the use of microservices architecture, different explanation microservices with their integrated knowledge bases, accessible only to them, can be used in such contexts and mitigate the problem.

Ontological Approach to Creating Hybrid Technology.

In implementing CDSS, a two-level ontological approach is used for knowledge base formation, characterized mainly by the separation of ontologies from knowledge bases. This separation allows for the scaling of CDSS to new diseases without altering the code of the knowledge base handler, as multiple knowledge bases can be developed based on a single ontology.

The scalability of CDSS is dependent on the generality of ontologies. General ontologies such as the ontology of disease diagnostics, treatment prescription (medication, rehabilitation, surgery), prognosis, and risk assessment of critical conditions, independent of medical specialty and disease group, have been created and utilized over the years to develop separate CDSS for diagnostics and treatment across various disease groups (respiratory, digestive, circulatory, genitourinary, musculoskeletal, nervous systems, endocrine system diseases, metabolic disorders, etc.).

Ontological knowledge models are implemented on the IACPaaS platform, which provides infrastructure for forming ontology-based knowledge portals and services. The ontological approach and semantic knowledge representation are human-understandable and machine-interpretable. Software solvers reason based on interpretable knowledge and save their results according to ontology explanations, allowing for the combination of functionality and disease groups.

Ontology of knowledge on Disease Diagnostic.

The ontology of knowledge on disease diagnostics provides the capability to form knowledge bases with the following properties.

A symptom complex is a set of patient signs (complaints, objective, laboratory, instrumental studies), or a syndrome (a group of signs united by a common pathogenesis). Symptom complexes (clinical picture) of diseases can be described considering user categories. Using categorization allows for the most accurate description of clinical manifestations and data from laboratory and instrumental indicators, for example, considering age, profession, conditions like pregnancy, etc., enhancing the informational significance of any symptom complex.

The value of each sign is extended by a range of modality values (necessity, characteristic, possibility, intensity, etc.);

Disease diagnostics can be represented by alternative symptom complexes, facilitating the formalization of various approaches to identifying reliable disease signs, allowing for the selection of the most sparing, rapid, or cost-effective diagnostic approach;

A diagnosis can be described considering etiology, pathogenesis, variant of progression, etc., for differential diagnosis of diseases and the selection of appropriate treatment methods;

For a disease, necessary conditions that predispose or contribute to its development can be set, such as age, gender, season, etc.

Diagnostics can take into account the values of characteristics altered by event impacts, such as ecological factors (e.g., polluted air, water, the impact of harmful industrial, agricultural, household, and other factors), quantitative and qualitative inadequacy of nutrition, disruption of the orderly and optimal balance of work and active rest, social factors (e.g., frequent conflict situations), and more.

In describing a disease, different variations in the dynamics of sign values can be outlined.

Ontology of Knowledge on Disease Treatment

The ontology of knowledge on disease treatment facilitates the formation of knowledge about the treatment of a specific disease or a group of diseases sharing common pathogenetic principles, etiological components, or clinically significant symptomatic manifestations. This disease ontology possesses the following characteristics.

Therapy Model: Logically coherent representations of the principles and extents of therapy for a given pathological process, encompassing Type, Goal, and Scheme of therapy. For a single disease, several alternative therapy models may be described (for example, in accordance with clinical guidelines from the Ministry of Health, new authorial treatment methods);

Type of Therapy: Encompasses a class of concepts directly describing the type of therapy, such as etiotropic, pathogenetic, symptomatic, empirical, and other types of therapy;

Goal of Therapy: A class of concepts characterizing the purpose of the treatment, such as hemostatic, antiemetic, antipruritic, detoxification, or mucolytic therapy. Characteristics that define the goal of therapy include clinical data descriptions that allow for the recognition of the achievement of therapy goals;

Key Element of Ontology: A complexly structured block of conditions accompanying each section of the ontology, enabling the formal representation of necessary clinical criteria that determine its application conditions in the treatment of a given disease;

Therapy Scheme: Defines the list of active substances, their combinations, administration regimen, and dosages of medicinal products (MPs) for optimal disease treatment. This section of the ontology is structured as follows: Condition for using this group of MPs, Group of alternatively used MPs, Complexly used MPs. The group of alternatively used MPs contains the following nodes: Active Substance and Jointly Used MPs. Each MP is described by a group of terms defining its clinical necessity: Prescription Variant, including elements: Dosage, Release Form, Application Method, Frequency of Application, Duration of Application.

Control Points: The Control Points are specified for each MP, for Therapy Effectiveness Assessment, Control of Expected Side Effects, Condition on the Active Substance, and Trade Names of the Active Substance. Control points for therapy effectiveness assessment allow for monitoring the application of the active substance, including descriptions of the disease characteristic/s for treatment monitoring. Control of expected side effects – a term of the ontology defining the safety of using MPs, includes a characteristic describing the type of side effect/adverse effect and the frequency of its monitoring.

The ontology of knowledge on prognosis and risks of conditions and diseases.

This ontology is designed for the formal description of the body's state dependency on the combination of observed signs and influencing factors, or pathology development variants depending on known factors. This ontology possesses the following properties.

In one knowledge base, alternative methodologies for assessing personalized risks can be described;

When describing risks, the type of threat, degree of threat, name/author of the methodology, possible necessary conditions, as well as the method of determination are described. The type of threat is characterized by: risk of occurrence, risk of first case, risk of recurrence, risk of progression to a severe stage, life expectancy prognosis, risk of death; the degree of threat is expressed as a percentage (probability) or a scale value (ranging from "very high risk" to "no risk");

The ontology subtree with the root 'Method of Determination' allows the integration of different methods for determining risks and state prognosis in one CDSS: calculated by a model, calculated by a formula, calculated by a declarative formula or determined by declarative knowledge).

Thus, the ontology on the prognosis and risks of states and diseases serves not only as a structure for describing declarative knowledge but also as an integrating structure, ensuring the fulfillment of the main requirement of CDSS - its hybridity.

3.3. Vision for Cloud-Edge Integration

As previously discussed, modern Clinical Decision Support Systems (CDSS) tend to be cloud-based and process data distributively, which definitely has its advantages and drawbacks (see Chapters 1 and 2). We believe that combining cloud computing with edge computing has enormous potential, as demonstrated in one of studies [6]. By edge level, we mean the level closest to the user, accessible without an internet connection, such as a local hospital network.

We have identified the following potential tasks and problems associated with the exclusive use of cloud computing.

The need for real-time data processing and the absence of a stable internet connection. For patients in intensive care and their physicians, instantaneous decision-making and data retrieval can be critical, with delays in data transmission or even communication breakdowns being a common occurrence with cloud computing alone. This situation worsens in remote areas or in meteorological phenomena like snow or sand storms.

Transmission of large data volumes and traffic costs. Roughly, a file containing 1 minute of HD endoscopy video (1920-1080px) can occupy 35-40 MB. For a 10-minute procedure, this already amounts to 350-400 MB, which, even with modern technologies, can pose a problem when uploading to the cloud, not to mention data transmission within the CDSS to the final processing service. Needless to say, the volume of such traffic generated by all patients in a medical institution over a year can become a significant expense item and a limiting factor in CDSS implementation, indicating that data preprocessing is an important stage.

Confidentiality. Compliance with regulations for the protection, storage, and processing of patient personal data becomes much more complex when these data are sent to a cloud controlled by a third party.

Interaction with wearable patient sensors and IoT devices. If such functionality is not integrated into the institution's HIS, or if the HIS is unable to communicate this data to the CDSS due to its own implementation limitations, processing signals will, one way or another, require local CDSS presence for interacting with these devices.

A solution to these problems could be the use of a distributed structure that employs both cloud and edge computing, thus combining them into cloud-edge. Edge computing could be carried out on a separate server, physically connected to the local network of the medical institution, but also part of the overall cloud infrastructure. Updating and servicing the CDSS portion located on this node should also be centralized and automated, as with cloud services. Depending on the end user's needs, this node can duplicate the functionality of all features available in the cloud or only selected ones. These could be special models, specifically developed for the institution and not available to other cloud users, or conversely, the most popular functions in everyday practice, access to which should not be limited in case of cloud connection issues.

It must be acknowledged that the edge node can have its drawbacks, making it unavailable for implementation for every user. The most serious of these is the requirement for technical equipment in medical institutions: namely, the availability of certain IT infrastructure, whose accessibility significantly exceeds that of cloud services. The presence of staff to maintain the network's functionality is also required. Additionally, the equipment needed to set up an edge node in the local network requires investments that may be unaffordable for small clinics in remote regions.

If a medical institution can afford it, then among the potential advantages of using a cloud-edge approach are: data preprocessing (such as images, videos, signals), reduced delays and increased timeliness of clinical decisions, simplification of data security compliance, autonomy, and continuous operation in conditions of unstable internet connection, and the ability to process signals from wearable patient sensors and IoT devices in real time.

In this chapter, we mentioned several diverse scenarios where, in our opinion, the implementation of an edge node could be beneficial. Currently, our team does not have the capability to conduct a thorough analysis and testing of each of these to describe in this study, as detailed examination of each case would require the volume of a separate publication. Nevertheless, we intend to test and implement some of the proposed scenarios in the future and evaluate the effectiveness of this implementation, both in terms of patient benefits and economic profitability and support for the medical institution.

3.4. Architecture

The hybrid technology, on which the CDSS is developed, features a distributed microservices architecture (see Fig. 1). The system's functionality is implemented through models (software computational agents) or knowledge bases and solvers (software solvers and knowledge bases for diagnostic/treatment/rehabilitation/screening protocols). Each CDSS task - whether it's the prognosis or diagnosis of a specific disease or complication - is handled by a separate microservice. To expand the functional tasks solved by the CDSS, new microservices need to be developed and integrated into the system following a standardized protocol, where the formalized description of the developed agent's interface is stored in the knowledge base.

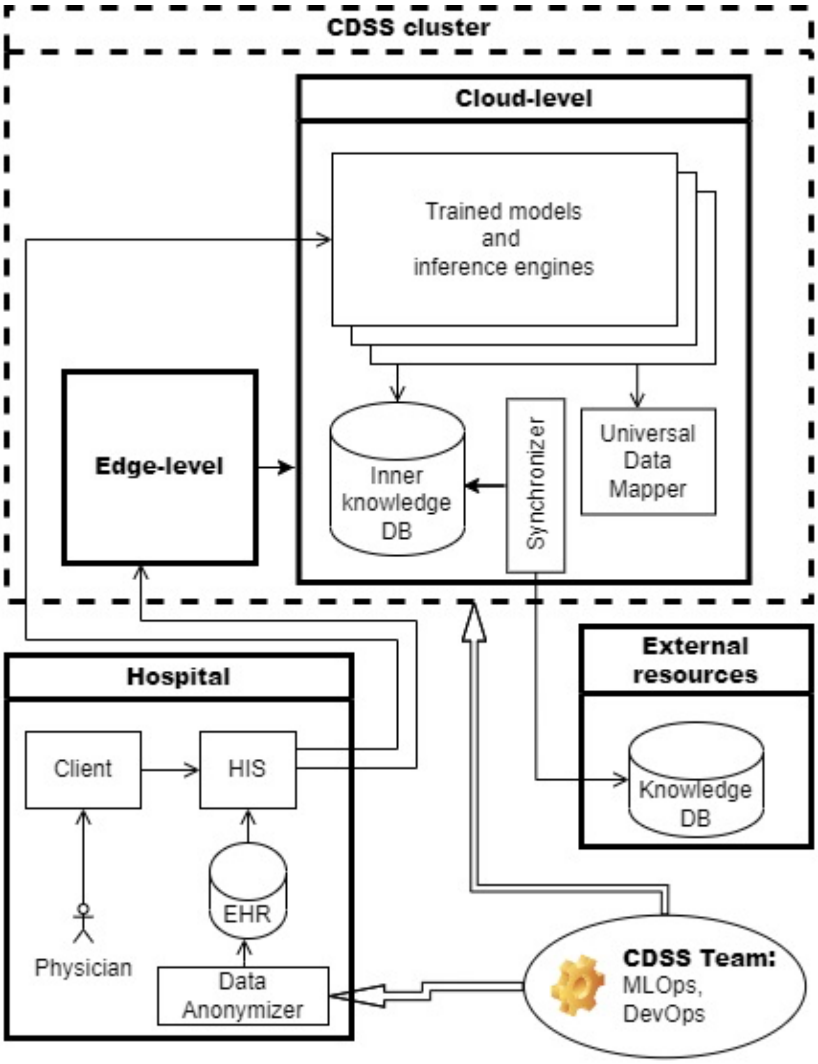

Figure 1. High-Level scheme of CDSS organization.

If the functionality of the CDSS is expanded by a new scale, which is calculated by the solver, then to expand the functionality, it is necessary to update the knowledge base with the corresponding scale's calculation rules. Since access to the software computational agents is via HTTP, the execution of a microservice is possible on different computational nodes, allowing the CDSS to be distributed. The integration of knowledge bases, ontological solvers, and machine learning models makes the CDSS hybrid. The main components of the CDSS are: (1) System Services, ensuring the solution of integration tasks with the HIS (Universal Data Mapper), routing of requests (Message Routing), including to computational agents and intelligent solvers; (2) Knowledge Bases of Clinical Medicine, describing formalized medical protocols for diagnostics, prevention, treatment, clinical recommendations, directories of medicinal products, terms, formalized explanations (interpretation) of prognostic models and the results of their application, etc.; (3) Computational Agents (Microservice), implementing prognostic or diagnostic models of machine learning, algorithms for calculating scores or the probability of developing a fatal or non-fatal event; (4) Intelligent Problem Solver, through which some conditions are checked and actions are implemented according to the ontological tree of the knowledge base; (5) HIS with EHR database; (6) EHR databases, including those on which the prognostic models were trained and databases of research, training, and testing of models.

3.5. Technologies

Developed with the above-described concept of hybrid architecture in mind, CDSS must integrate various components distributed in the IT space. Therefore, the system has a primary method of message and data exchange - the Https protocol and JSON exchange format. The task of integrating CDSS with HIS can be accomplished in several ways. If HIS supports functionality extensions through a procedure initiated by a physician, this procedure must ensure the calling of HIS's server part and transmitting the message to CDSS through it. The technologies used for this are determined by the HIS being used. For managing knowledge bases, the IACPaaS cloud platform is used [21, 22]. The IACPaaS platform allows the creation of formalized knowledge bases, describing prognostic and diagnostic models, their interpretation, clinical recommendations, explanations, etc. The interaction of microservices is organized using the Kubernetes orchestrator (Fig.2).

For brevity, only interaction with one medical institution is presented, but the system is designed for simultaneous interaction with multiple clients using different HIS and geographically distant from each other. In the diagram, the pods for the Edge node are shown in a simplified form (without highlighting microservices), and for the Worker node - in full.

The main elements of the CDSS architecture at the low level are:

Kubernetes cluster, including three typical nodes (nodes): Master - responsible for cluster configuration (contains control plane, etcd, and other standard Kubernetes environment components) and provides the main API Gateway for accessing CDSS; Worker - one or more identical nodes where microservices are placed in cells (pods) implementing machine learning prognostic models. The cluster must have at least one worker node; Edge - a variant of the worker node but located not in the cloud, but in the local network of a medical institution with its HIS.

HIS server and client are implemented in the space of the medical institution.

The IACPaaS server manages knowledge bases and includes a solver that ensures the traversal of the tree of the formalized knowledge base. If necessary, cluster services can access external knowledge bases, databases, or services.

Important features of the Kubernetes orchestrator include ease of deployment and maintaining the functionality of containers with microservices. Most functions are managed by the control plane, located on the Master node. The control plane's functions include: managing the API server (API server provides a REST API for managing Kubernetes objects such as pods, services, and controllers), scaling and managing pods, managing nodes, configuration management, access management (including user and service authentication and authorization), and network management (Control plane allows managing the network in the Kubernetes cluster, including setting up and managing services and access to them), monitoring, and logging. To ensure CDSS performance, the Kubernetes tool - Ingress, can be used, allowing traffic routing and load balancing in the cluster, enabling the solution to scale with an increasing number of requests. Another way to increase performance is cached computations, stored outside containers in Persistent Volume (PV). PV is an abstraction mechanism allowing to allocate some data storage space outside the pod but make it accessible within the pod. The cache of computations and the updated knowledge base placed in PV will not depend on the pod's lifecycle.

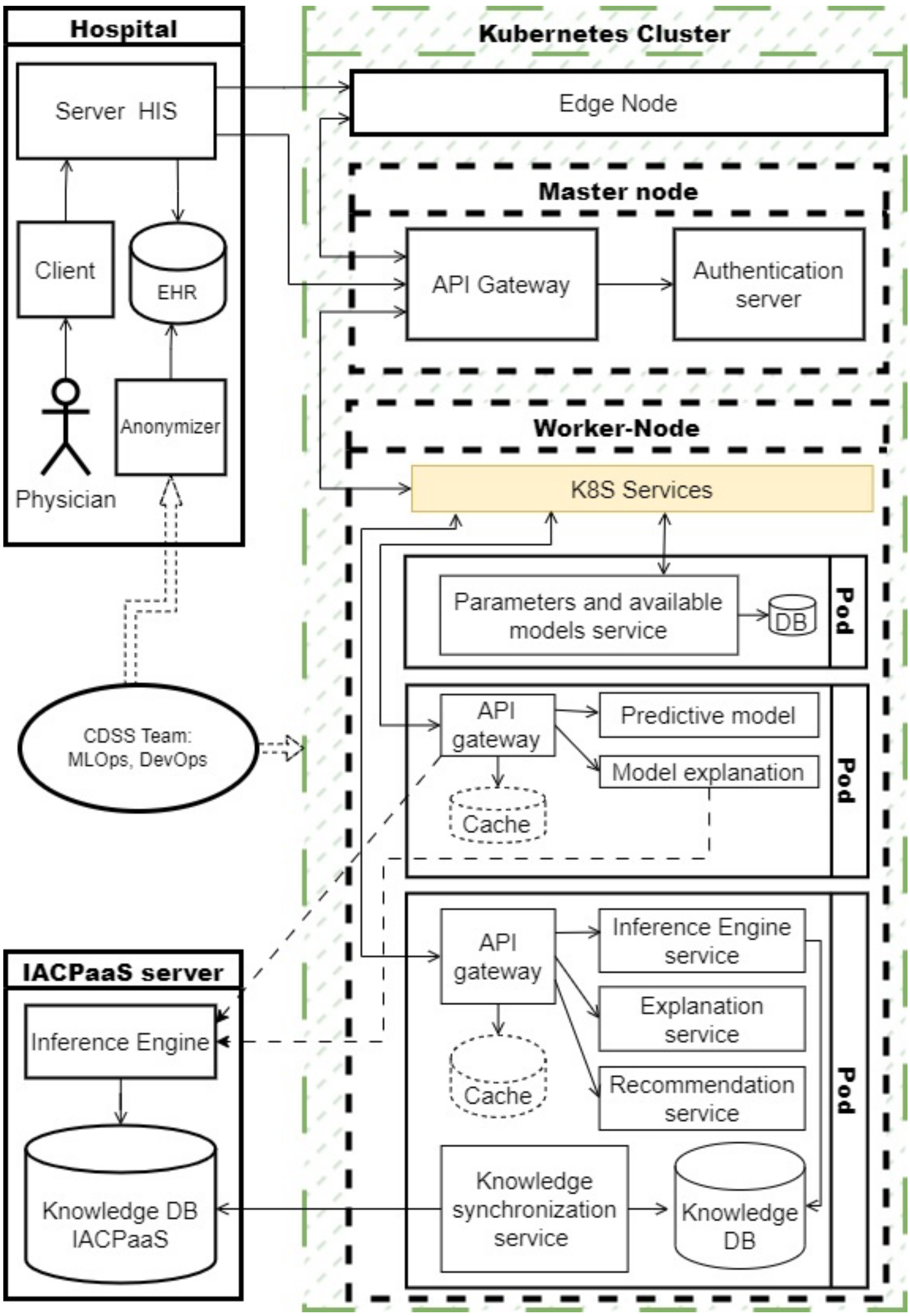

Figure 2. Low-level scheme of CDSS organization.

Each prognostic model integrated into the CDSS is implemented through a separate microservice on the worker-node. And since business logic requires not only computations but also interpretations of the obtained values, forming recommendations, and caching, the microservices responsible for these tasks are combined into pods. An unlimited number of nodes and pods with microservices, which are automatically maintained in a working state by the Kubernetes orchestrator, can be deployed within the cluster, ensuring the transparent scaling of the CDSS. Load balancing between pods and microservices on different nodes is done using the Ingress mechanism mentioned above. Scaling limitations are imposed by available resources (physical servers and computational power). However, increasing computing performance with simultaneous access by several clients, as well as increasing fault tolerance, can be critical in emergency medical practice situations. External request authorization to microservices in Kubernetes is organized using: Kubernetes RBAC (Role-Based Access Control), Istio, API Gateway, OAuth2.

One of the features of the CDSS is the possibility to deploy it not only in the public cloud but also within the network of a medical institution, then all computations will be located in the Edge-node. The same microservice can exist in several instances with different IP addresses on one or more nodes, and the proxy service allows load balancing on the fly according to the round-robin principle between instances. Such configuration offers several significant advantages, the main ones being: increased data exchange speed (especially relevant for image processing), priority use of computational power (the node can be configured to process requests only from one HIS), increased fault tolerance in conditions of unstable internet connection (local network remains operational). The capabilities of Kubernetes node configuration allow making proxy services (K8S Services) available in the local network, where the physical server of the Edge node and the HIS server are located.

CDSS microservices are grouped into cells. The following typical cells are assumed:

Informational - informs the client about available prognostic models and their parameters, including predictors, conditions of application, location, rules of result interpretation, clinical recommendations, etc.;

Transformation - transforms data received from HIS into a data representation format for the predictive model and the model's work result into a format understandable by HIS;

Service implementing the predictive model (including computation, model explanation, and recommendations);

Routing service for distributed computations. Preliminary data processing, transformations, and forecast or diagnosis calculations can be carried out both on the Edge-node within the network of a medical institution and in the public cloud. In some cases, this can be done automatically or be a condition of the agreement with the medical institution;

The solver cell includes the solver service, services for finding explanations and recommendations. Service for synchronizing the knowledge base with an external source. It's important to note that the knowledge base and cache should be placed in a Persistent Volume to avoid losing relevance when restarting the pod.

4. IMPLEMENTATION

The development of the CDSS is being carried out for the Medical Center of the Far Eastern Federal University (FEFU). Several evolutionary stages were designated for the implementation of the CDSS. The main task of the first stage was to develop an MVP (Minimum Viable Product). At this stage, we do not use the Kubernetes orchestrator but focus on creating a viable system based on a microservices architecture. CDSS services are hosted on the FEFU server. The interaction services with CDSS are integrated into the HIS "1C Hospital". In the electronic medical record, a doctor can select available prognostic models. The code of the medical history/ambulatory card in the 1C HIS of the FEFU Medical Center is used by the CDSS as the primary key for a record about a patient within one episode for medical histories or about one patient - for an ambulatory card. Being a primary key - it must be unique for each data set. If an already existing code is transmitted to the CDSS services, the data is overwritten. Subsequently, the record can be found and processed by the user through the web interface and will be displayed in the journal.

Several prognostic models have been implemented as microservices, some of which are widely used in clinical cardiology and cardiothoracic surgery (SCORE, SCORE 2, GRACE, EuroSCORE II, Diamond-Forrester, etc.), as well as proprietary machine learning models developed by the authors, implementing tasks such as predicting in-hospital mortality after cardiac surgery - coronary bypass and percutaneous coronary intervention, development of postoperative atrial fibrillation, and pre-test assessment of the likelihood of obstructive coronary artery disease.

Each service model is represented in the knowledge base by several fields: service name, input and output parameters; normative values of input indicators (if available); URL link to the service; interpretations and a set of recommendations for doctors and patients, depending on the obtained results.

For FEFU Medical Center specialists, a doctor's workstation and electronic documents of 1C HIS have been developed, where after the calculation is performed using prognostic models, the CDSS returns the results of the models, the interpretation of these results, and recommendations for reducing the risks of adverse events for this patient.

A general example of system application might be the following algorithm:

The HIS server, upon the doctor's request or regularly (using an agent), authenticates through the API Gateway in the CDSS and gets the current list of prognostic models available to the medical institution and doctor, as well as their parameters (predictors).

The doctor in HIS selects the necessary model from the available ones and makes a request for a forecast or diagnosis assessment.

The HIS server selects the necessary data (predictors) for the model's calculations from the EHR database.

If HIS has access to the Edge-node, the request with data is sent to the local predictive model microservice located on this node, otherwise - the request is sent to the cloud, through the API Gateway, to one of the available Worker nodes, where the corresponding microservice is implemented.

The external interface of the model microservice (Model gateway service) uses the parameter display service to transform data from EHR into the input parameters needed for the model's calculation.

In the CDSS, there is a cache that saves the results of the model calculation and provides them upon a repeated request.

After calculating the forecast result, explanations and recommendations corresponding to the calculations are added to the response. The choice of interpretations and explanations corresponding to the result is performed by the solver of the intelligent knowledge base.

The model's response, if necessary, passes through the display service for transformation into a format understandable by HIS and, together with other data, is returned to HIS.

Based on this algorithm, we propose a possible scenario for everyday practice.

Scenario: A patient with suspected coronary heart disease visits a cardiologist at the FEFU Medical Center. The physician decides to use the CDSS to assess the risk and determine the optimal treatment strategy.

Data Request: The physician in HIS selects the SCORE 2 model to assess the patient's condition. After selecting the model, the HIS server automatically collects the necessary patient data from the EHR.

Processing the Request: Since an edge-node is available in the medical center, the request with data is sent to the local SCORE 2 microservice installed on this node.

Forecast Calculation: The model microservice processes the data, using the external interface to transform them into the required format. The calculation results are cached to increase the efficiency of future requests.

Interpretation and Recommendations: The obtained forecast is accompanied by explanations and recommendations prepared by the solver of the intelligent knowledge base of the CDSS.

Providing Results to the physician: The results, along with interpretations and recommendations, are transmitted back to HIS, where they are displayed in a format understandable to the doctor.

Application of Recommendations and Decisions: Based on his own knowledge and the decision proposed by the CDSS, which matches the physician's expectations, the doctor confidently diagnoses the disease in the patient and prescribes the most effective treatment, in his opinion.

Application Results: The treatment proves effective, and the patient's condition stabilizes.

This slightly naive scenario is intended to express our hope that future research on CDSS integration will demonstrate rapid and accurate risk assessment and treatment recommendations, the improvement of the timeliness and quality of medical care, the reduction of the likelihood of errors and improvement of the outcomes for patients.

5. CHALLENGES AND FUTURE DIRECTIONS

In the course of developing the CDSS for the Medical Center, we encountered a number of technical and clinical challenges that prompted research into improving the architecture of the CDSS. The key aspects are: more accurate forecasting; better explanation of decisions and recommendations; organization of work of several teams conducting parallel development, implementation, and testing of new machine learning models; ensuring data security and confidentiality; and connecting new medical institutions to the system.

The lack of unified standards for HIS creates difficulties in integrating CDSS with existing solutions in medical institutions. The full implementation of a service to translate requests from HIS to a microservice remains a prospective task.

Scalability and performance of our system also require attention. At present, the number of users is minimal, as is the number of patients. We are confident that the scalability we are building into the CDSS will cope with the increasing load, but testing in real-life conditions remains ahead.

From the clinical practice perspective, an important aspect is improving the accuracy and reliability of prognostic models. Another task is working with medical staff: training in the use of CDSS, gathering feedback, improving the interface, evaluating the understandability and acceptability of the proposed explanations, their impact on decisions made by doctors and medical staff, and most importantly - the benefits for patients.

We are focusing on developing more sophisticated methods of explaining prognostic models in medicine in general, and cardiovascular diseases in particular. As mentioned earlier, we are also facing requests for implementing AI and XAI methods in other everyday diagnostic tasks, whose effective resolution requires a high-quality implementation of cloud-edge computing.

6. CONCLUSION

In this study, the authors proposed the concept and architecture of a hybrid CDSS, integrating machine learning models and intelligent knowledge bases that describe methods for assessing cardiovascular risks, their explanations, and recommendations for limitation. The microservice approach provides flexibility, scalability, and independence from the used hardware resources. The application of Edge computing optimizes data processing and reduces server load. The use of cloud technologies and containerization of services ensures the system's rapid adaptation to the changing conditions and needs of medical institutions. Formalized knowledge bases with a synonym mechanism help solve the issues of using various terms in HIS. The system is being developed using an iterative approach. Future stages include expanding the available pool of prognostic models, enhancing the architecture through the use of orchestrators, and integration with new HIS. The development experience of this system underscores the necessity of an interdisciplinary approach based on the cooperation of specialists in information technology, machine learning, clinical medicine, and medical institution management. This is a key factor in creating innovative solutions for implementing digital medicine projects.

7. DISCUSSION

In this research, we focused on the development of the hybrid CDSS architecture for the Medical Center of the Far Eastern Federal University. Special attention was given to XAI methods, where hybridity plays a key role, as well as the use of a microservice architecture with Edge computing nodes, and exploring the potential opened by this approach. This innovative construction not only ensures flexibility and scalability but also lays the foundation for future implementation of technologies such as Kubernetes to improve service management and deployment.

It's important to note that at this stage, our system is in its initial implementation phase - more of a prototype than a fully-fledged system. We continue to work on collecting and analyzing data to evaluate the effectiveness of using the prototype in the FEFU Medical Center. These data will help us improve the system and adapt it to the real conditions of clinical practice.

Currently, our main focus is on refining the CDSS architecture, including planning the implementation of an orchestrator and further development of microservices. As the functionality expands and the number of integrated machine learning models increases, we expect significant improvement in system performance and capabilities.

In conclusion, the use of the existing CDSS prototype in the FEFU Medical Center and research into potential directions in architecture design confirms the importance of hybrid systems and microservice architecture in modern digital medicine.

Focusing primarily on cardiovascular diseases, our CDSS provides an opportunity for teams of scientists working in other areas of medical diagnostics to join. This project becomes an example of how technological innovations can contribute to the development of the medical industry, improve the quality of medical care, and increase the efficiency of clinical decisions.

Declaration of Competing Interest. The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments. Funding: The research was conducted with the financial support of the grant from the Russian Science Foundation (RSF) # 23-21-00250 https://rscf.ru/project/23-21-00250/.

REFERENCES

[1] "Arden Syntax for Medical Logic Systems Standard Version 2.5." American National Standards Institute, Washington, DC (2004).

[2] Acosta, J.N., Falcone, G.J., Rajpurkar, P., et al. "Multimodal biomedical AI." Nat. Med. 28, 1773-1784 (2022). doi: 10.1038/s41591-022-01981-2.

[3] Ahsan, M.M., Luna, S.A., Siddique, Z. "Machine-Learning-Based Disease Diagnosis: A Comprehensive Review." Healthcare (Basel) 10(3), 541 (2022). doi: 10.3390/healthcare10030541.

[4] Amann, J., et al. "Explainability for artificial intelligence in healthcare: a multidisciplinary perspective." BMC Med. Inform. Decis. Mak. 20, 1, 310 (2020). doi:10.1186/s12911-020-01332-6.

[5] Antoniadi, A.M., Du, Y., Guendouz, Y., Wei, L., Mazo, C., Becker, B.A., Mooney, C. "Current challenges and future opportunities for XAI in machine learning-based clinical decision support systems: a systematic review." Appl. Sci. 11(11), 5088 (2021).

[6] Babic, B., et al. "Beware explanations from AI in health care." Science 373, 284-286 (2021). DOI:10.1126/science.abg1834.

[7] Berner, E.S. "Clinical Decision Support Systems." Springer (2016).

[8] Boehm, K.M., Khosravi, P., Vanguri, R., et al. "Harnessing multimodal data integration to advance precision oncology." Nat. Rev. Cancer 22, 114-126 (2022). doi: 10.1038/s41568-021-00408-3.
[9] Bolourani, S., Tayebi, M.A., Diao, L., Wang, P., et al. "Using machine learning to predict early readmission following esophagectomy." J. Thorac. Cardiovasc. Surg. 161(6), 1926-1939.e8 (2021).
[10] Chang, A.C. "Intelligence-Based Medicine. Artificial Intelligence and Human Cognition in Clinical Medicine and Healthcare." Chapter 5 - Machine and Deep Learning, pp. 67-140 (2020).
[11] Changyu, D., Ji, X., et al. "Integrating Machine Learning with Human Knowledge." DOI: 10.1016/j.isci.2020.101656.
[12] Committee on Diagnostic Error in Health Care, et al. "Improving Diagnosis in Health Care." Ed. by E.P. Balogh et. al., National Academies Press (US), 2015. doi:10.17226/21794.
[13] Cremonesi, F., Planat, V., et al. "The need for multimodal health data modeling: A practical approach for a federated-learning healthcare platform." J. Biomed. Inform. 2023, 141, 104338. doi: 10.1016/j.jbi.2023.104338.
[14] Ding, S., Li, L., Li, Z., Wang, H., Zhang, Y. "Smart electronic gastroscope system using a cloud-edge collaborative framework." Future Gener. Comput. Syst. 100, 395-407 (2019).
[15] Dragoni, N., et al. (2017). "Microservices: Yesterday, Today, and Tomorrow." In: Mazzara, M., Meyer, B. (eds) Present and Ulterior Software Engineering. Springer, Cham. doi: 10.1007/978-3-319-67425-4_12.
[16] Du, Y., McNestry, C., Wei, L., Antoniadi, A.M., McAuliffe, F.M., Mooney, C. "Machine learning-based clinical decision support systems for pregnancy care: A systematic review." Int. J. Med. Inform. 173, 105040 (2023).
[17] Durán, J.M. “Dissecting scientific explanation in AI (sXAI): A case for medicine and healthcare.” Artif. Intell. 297, 2021, 103498. doi:10.1016/j.artint.2021.103498.
[18] El-Sappagh, S.H., El-Masri, S. "A distributed clinical decision support system architecture." J. King Saud Univ. - Comput. Inf. Sci. 26(1), 69-78 (2014).
[19] Evans, R.S., Pestotnik, S.L., Classen, D.C., et al. "A computer-assisted management program for antibiotics and other antiinfective agents." N. Engl. J. Med. 338(4), 232-238 (1998).
[20] Grampurohit, S., Sagarnal, C. "Disease Prediction using Machine Learning Algorithms." 2020 Int. Conf. for Emerging Tech. (INCET), Belgaum, India, 2020, pp. 1-7. doi: 10.1109/INCET49848.2020.9154130.
[21] Gribova, V., Moskalenko, P., Timchenko, V., Shalfeyeva, E. "Intelligent Services Development Technology Using the IACPaaS Cloud Platform." Commun. Comput. Inf. Sci. 1625, 19-38. Springer, Cham (2022).
[22] Gribova, V., Shalfeeva, E. "Ontology of anomalous processes diagnosis." Int. J. Intell. Syst. 36(1), 291-312 (2021).
[23] Holzinger, A., Biemann, C., Pattichis, C.S., Kell, D.B. "What do we need to build explainable AI systems for the medical domain?" (2017).
[24] Huang, Y., Li, J., Li, M., et al. "Application of machine learning in predicting survival outcomes involving real-world data: a scoping review." BMC Med. Res. Methodol. 23, 268 (2023). doi:10.1186/s12874-023-02078-1.
[25] Irakli, N., Mitra, R., McLarty, M., and Amundsen, M. "Microservice Architecture: Aligning Principles, Practices, and Culture." O'Reilly Media, Inc., 2016. ISBN:978-1-4919-5625-0.
[26] Iscra, K., Miladinović, A., Accardo, A. "Interpretable machine learning models to support differential diagnosis between Ischemic Heart Disease and Dilated Cardiomyopathy." Procedia Comput. Sci. (2022).
[27] Kawamoto, K., Lobach, D.F. "Design, implementation, use, and preliminary evaluation of SEBASTIAN, a standards-based web service for clinical decision support." Proc. AMIA Symp. (2005).
[28] Kline, A., Wang, H., Li, Y., et al. "Multimodal machine learning in precision health: A scoping review." npj Digit. Med. 5, 171 (2022). doi: 10.1038/s41746-022-00712-8.
[29] Kobrinsky, B.A. "Medical Informatics." Moscow: Academy, 2009. ISBN 978-5-7695-5442-1.
[30] Kovalchuk, S.V., Kopanitsa, G.D., Derevitskii, I.V., Matveev, G.A., Savitskaya, D.A. "Three-stage intelligent support of clinical decision making for higher trust, validity, and explainability." J. Biomed. Inform. 127, 104013 (2022).
[31] Kundu, S. "AI in medicine must be explainable." Nat. Med. 27, 1328 (2021). doi: 10.1038/s41591-021-01461-z.
[32] Markus, A.F., Kors, J.A., Rijnbeek, P.R. "The role of explainability in creating trustworthy artificial intelligence for health care: A comprehensive survey." J. Biomed. Inform. 113, 103655 (2021). doi:10.1016/j.jbi.2020.103655.

[33] Nadkarni, P., Miller, R. "Service-oriented architecture in medical software: promises and perils." J. Am. Med. Inform. Assoc. 14(2) (2007).
[34] Osmolovskiy, I.S., Zarubina, T.V., Shostak, et al. "Development of a knowledge base structure in the field of gout diagnosis." Siberian J. Clin. Exp. Med. 37(3), 2022.
[35] Oyenuga, S.O., Garg, L., Bhardwaj, A.K., Shrivastava, D.P. "Cloud-Based Clinical Decision Support System." Conf. Proc. ICDLAIR2019, pp. 220–234.
[36] Panigrahi, N., Ayus, I., Jena, O.P. "An Expert System-Based Clinical Decision Support System for Hepatitis-B Prediction & Diagnosis." Machine Learning for Healthcare Applications (2021).
[37] Peixoto, H., Guimarães, T., Santos, M.F. "A New Architecture for Intelligent Clinical Decision Support for Intensive Medicine." Procedia Comput. Sci. 170, 1035-1040 (2020).
[38] Peleg, M. "Computer-interpretable clinical guidelines: A methodological review." J. Biomed. Inform. 2013, 46(4), 744-763; Young, O., Shahar, Y., et al. "Runtime application of Hybrid-Asbru clinical guidelines." J. Biomed. Inform. 2007, 40(5), 507-26.
[39] Quazi, S. "Artificial intelligence and machine learning in precision and genomic medicine." Med. Oncol. 2022, 39(8), 120. doi: 10.1007/s12032-022-01711-1.
[40] Ram, P., Berg, D., Tu, S., et al. "Executing clinical practice guidelines using the SAGE execution engine." Medinfo 11(Pt 1), 251-255 (2004).
[41] Semenov, I., Osenev, R., Gerasimov, S., Kopanitsa, G., Denisov, D., Andreychuk, Y. "Experience in Developing an FHIR Medical Data Management Platform to Provide Clinical Decision Support." Int. J. Environ. Res. Public Health. (2020).
[42] Shamout, F.E. "Machine Learning for Decision Support Systems: Prediction of Clinical Deterioration." In: Rivas, H., Boillat, T. (eds) Digital Health. Health Informatics. Springer, Cham, 2023. doi:10.1007/978-3-031-17666-1_7.
[43] Smiti, A. "When machine learning meets medical world: current status and future challenges." Comput. Sci. Rev. 37, 100280 (2020).
[44] Sutton, R.T., Pincock, D., Baumgart, D.C., Sadowski, D.C., Fedorak, R.N., Kroeker, K.I. "An overview of clinical decision support systems: benefits, risks, and strategies for success." npj Digit. Med. 3(1), 17, (2020).
[45] Tarnowska, K.A., Dispoto, B.C., Conragan, J. "Explainable AI-based clinical decision support system for hearing disorders." AMIA Jt Summits Transl Sci Proc. 595–604 (2021).
[46] Wright, A., Sittig, D.F. "A four-phase model of the evolution of clinical decision support architectures." Int. J. Med. Inform. 77(10), 641–649 (2008).
[47] Xiaojun, C., Jia, S., Xiang, Y. "A review: Knowledge reasoning over knowledge graph." Expert Syst. Appl. 2020, V. 141. doi.org/10.1016/j.eswa.2019.112948.
[48] Zaki, J., Islam, S.M.R., Alghamdi, N.S., Abdullah-Al-Wadud, M., Kwak, K.-S. "Introducing Cloud-Assisted Micro-Service-Based Software Development Framework for Healthcare Systems." IEEE Access, vol. 10, pp. 33332-33348, 2022. doi: 10.1109/ACCESS.2022.3161455.