Development of a Rule-Based Knowledge System for Lung Cancer Diagnosis

Branislav Stojković[1][0009-0008-4219-6597], Dragana Filipović[1][0000-0003-1921-1732], Anđela Trajković[1][0000-0002-0223-1756] and Siniša Nikolić[1][0000-0002-2104-0373]

[1] Faculty of Technical Sciences, The University of Novi Sad, Novi Sad, Serbia
stojkovic.sv64.2020@uns.ac.rs
dragana.filipovic@uns.ac.rs
trajkovic.andjela@uns.ac.rs
sinisa_nikolic@uns.ac.rs

Abstract. Lung cancer remains a formidable challenge in modern medicine, characterized by low survival rates and high risks for patients. Improving the medicine by automating the part of the job is crucial for medical employees so they can focus on patient treatment. This paper introduces a clinical decision support system designed to match lung cancer symptoms with potential therapies by employing a knowledge base of rules to analyze patient symptoms and recommend personalized treatment options. The system continuously monitors patient data for symptoms before surgery, enabling early detection of critical conditions. This enables medical professionals to intervene promptly and make necessary adjustments to the treatment plan. In some cases, if the patient's condition deteriorates, the system can suggest postponing or canceling scheduled procedures to prevent complications. Moreover, the rule-based architecture ensures that the system remains flexible and up to date. New medical findings and treatment protocols can be easily integrated, allowing healthcare providers to access the most relevant and accurate information. This adaptability enhances decision-making and improves patient outcomes.

Keywords: Expert system, Drools, rule-based system, therapy, lung-cancer.

1 Introduction

Lung cancer is a highly challenging and often fatal disease [1]. Effective treatment requires accuracy, rapid response to changing symptoms, and careful management of critical conditions. The complexity of lung cancer necessitates a comprehensive understanding of diverse clinical, genetic, and pathological factors, along with the patient's overall health [2]. Successful treatment hinges on selecting the most appropriate therapy and diligently monitoring its progress.

Currently, lung cancer classification relies on the TNM system (Tumor, Node, Metastasis) [3]. However, the rising number of patients underscores the critical need for personalized therapy to improve treatment outcomes and effectively manage this disease on a larger scale. Achieving timely and accurate classifications in this context demands advanced technological support and expert systems [4].

This paper introduces a clinical decision support system for lung cancer diagnosis and treatment planning. This system assists medical professionals by:

- **Integrating clinical data:** The system compiles patient data from various examinations and utilizes TNM classification to determine the stage of cancer.
- **Recommending personalized treatment:** By considering the cancer stage, patient age, and overall health, the system generates tailored treatment recommendations.
- **Monitoring patient condition:** The system provides real-time monitoring of the patient's condition, particularly during the preoperative phase, issuing alerts for critical situations that may require adjustments to the treatment plan.

Developed as a web application accessible through any internet browser, the system is designed for both patients and oncologists. It employs Drools as its rule engine, enabling dynamic decision-making based on predefined rules and patient-specific data. Built on React, Spring Boot, and MySQL, the system ensures scalability, responsiveness, and robust data management.

The following section defines the research questions and methodology, while the third section provides an overview of related works mentioned in research papers and analysis of existing software solutions for cancer care. The software solution design is described in the fourth chapter, and the specific details of the implementation are in the fifth chapter. The last chapter discusses systems improvements and possible future work.

2 Key Research Questions and Methodology

The research in this paper seeks to address two key research questions:

- Is it practicable to create a clinical decision support system for lung cancer patients by relying on personalized patient data?
- If so, can we make therapy recommendations using a rule-based system where domain knowledge is stored in the form of formally defined rules?

This research aims to explore the feasibility of developing a rule-based system for personalized therapy recommendations in lung cancer treatment. It investigates how such systems can enhance decision-making in medical diagnostics. To achieve this, a structured methodology is applied, encompassing various key steps:

- **Literature Review:** Examining existing research on expert systems in medicine, particularly those focused on lung cancer diagnosis and treatment.
- **Solution Analysis:** Evaluating current solutions and software platforms used in medical diagnostics and treatment management.
- **Knowledge Acquisition:** Gathering expert knowledge from oncologists and medical sources to inform the system's rules and logic.

- **Rule Definition:** Establishing a comprehensive knowledge base that encompasses TNM classification, clinical parameters, genetic data, and other relevant information for personalized diagnosis and treatment.
- **Technology Selection:** Choosing and integrating appropriate rule-based systems and web technologies to build a functional and user-friendly application.

This paper further explores the system's design, implementation, and potential for future enhancements, including the integration of knowledge-based system for improved diagnostic accuracy and treatment personalization.

3 Related work

This section examines existing research and solutions related to clinical decision support systems, expert system in medicine, and applications for cancer care.

3.1 Expert System in Medicine

Expert systems, pioneered in the 1960s, aim to replicate human reasoning within a specific domain [5]. Unlike traditional programs that follow step-by-step procedures, expert systems utilize a knowledge base of domain-specific facts and rules to solve problems [6]. Several studies highlight their application in medicine:

- **Diagnosis and Treatment:** Research demonstrates the use of expert systems for diagnosis based on structured queries, such as yes/no questions, leading to treatment recommendations. This approach is particularly valuable for interactive medical applications requiring rapid and accurate diagnoses [7].
- **Chronic Disease Management:** Studies explore rule-based systems, often employing Drools, for managing chronic conditions like diabetes. These systems assist healthcare professionals in decision-making, potentially improving clinical outcomes and reducing errors [8].
- **Anesthesia Dosage:** Expert systems are used to determine anesthetic dosages during procedures. These systems consider patient parameters like age, weight, and medical history, and can generate alarms for adverse reactions. This application enhances patient safety by minimizing risks during interventions [9].

3.2 Analysis of Existing Cancer Care Solutions

Two prominent applications for cancer care are analyzed:

- **Outcomes4Me** [10]**:** This app empowers cancer patients to self-manage their treatment and access personalized medical information. It offers features like treatment recommendations based on latest guidelines, progress tracking, and health metric logging. Using advanced features is supported through a premium subscription.

- **Cancer.Net** [11]**:** This application supports cancer patients and families by providing comprehensive medical information and tools. Key features include symptom tracking, appointment scheduling, and educational resources. The application enables advanced features like detailed data analysis through its subscribed version.

3.3 Comparison and Differentiation

While both Outcomes4Me and Cancer.Net offer valuable resources for cancer patients, this study's proposed system differentiates itself in several ways:

- **Focus on Healthcare Professionals:** The system receives data through its dedicated endpoints and integrates with third-party applications. It supports healthcare providers with real-time monitoring and alerting mechanisms, ensuring timely intervention during the preoperative phase. Additionally, doctors can expand and refine the rule base, enhancing the system's accuracy and adaptability over time.
- **Open Access:** Unlike Outcomes4Me's premium model, the proposed system provides unrestricted functionality to all users.
- **Dynamic Treatment Planning:** The system allows for dynamic adjustment of treatment plans based on real-time patient data and monitoring.

4 System design

The proposed lung cancer diagnosis and treatment planning system streamlines automated decision-making during the preoperative phase. It integrates medical expertise and TNM classification standards to ensure the credibility and accuracy of its recommendations. Key design features include:

- **Comprehensive Data Management:** Cloud-based platform for input, processing, and analysis of patient data, including medical results, symptoms, and diagnoses.
- **Dedicated endpoints:** Dedicated endpoints are available for real-time data acquisition for vital signs like temperature, heart rate, etc.
- **User Roles:** Provides support for unregistered users, patients who receive personalized recommendations, and physicians who can access and manage patient data and treatment plans.
- **Data Modeling:** Utilizes 15 classes and 7 enumerations to organize complex medical data.
- **Modular Architecture:** Comprises a client application (React and TypeScript), a server application (Spring Boot), and a rule-based system (Drools) the extension of the system functionality.

4.1 Client application

The client application, built with React.js and TypeScript, operates as an interface that interacts with the integrated server application layer. It relies on technologies like Redux for state management, ensuring efficient data flow and updates. Axios is used for handling HTTP requests, including authentication and error management. Custom Hooks simplify state logic by modularizing reusable functionalities. These technologies work together to create a seamless, scalable, and maintainable application structure.

4.2 Server application

The server application uses the Spring Boot framework and is organized into three Maven projects: model, service and kjar. The project model contains all the entities and models used to store data in the database. It is organized into multiple folders, including models for views, therapies, users, alarms, and templates for generating rules. This structure enables a clear division of functionality and facilitates system maintenance and expansion.

The service project includes controllers, repositories and service classes. The controller folder manages HTTP requests and serves as a link between the client and server side, while the repository contains interfaces for working with the database. The mapper folder is used to map data between entities and DTOs, thus enabling a clear separation of application layers. Custom exceptions folder facilitates error handling and ensures better system stability.

5 System implementation

Drools was chosen to implement the rule-based system due to its compatibility with the Java ecosystem and its flexibility in creating complex rules. The rules are organized into several categories: forward chaining, CEP (Complex Event Processing), report generation rules, and template-based rules. This organization ensures easy maintenance and extensibility of the system.

Forward chaining, in particular, plays a critical role in the system's functionality. It allows the system to automatically infer conclusions and update patient data based on new information. For example, when patient data such as vital signs or test results are entered, forward chaining rules automatically update the patient's condition, providing medical professionals with immediate insight into their health. This proactive approach improves data management and simplifies patient progress tracking.

The system uses CEP to monitor real-time data, such as patient vital signs. Rules for scheduling medical examinations, such as CT scans, are activated based on specific criteria, such as the number of critical symptoms. For example, if a patient shows more than three critical symptoms, the system automatically schedules diagnostic examinations to ensure a timely response.

Source code is available on GitHub and a short video demonstrating its usage is available on [12]

5.1 Automatic update of patient status

The rules in the system enable automatic updating of data on the patient's condition, ensuring that medical professionals have real-time insights. Patient data is gathered through a manual input by healthcare personnel and automated integration with drools system. For example, when a CT or X-ray scan is completed, the results can be entered manually by medical staff and fed into the system via integrations with drools rule-based system. If abnormalities are detected, flags like ctDone and ctPositive are automatically set (Fig. 1**Error! Reference source not found.**). This process reduces manual intervention and enhances efficiency in tracking patient progress. Similarly, biopsy results that indicate malignant changes trigger an automatic update, setting the biopsyDone and biopsyPositive flags.

Beyond imaging tests, the system also handles various diagnostic procedures like: spirometry, spiroergometry, DLCO, and genetic tests. If any of these examinations yield concerning results, corresponding flags are updated to reflect the findings. This automated logic streamlines decision making, helping doctors promptly assess whether further intervention is necessary.

The rule-based system ensures seamless data flow, centralizing patient information in a structured and standardized manner. By integrating a knowledge-driven approach, the system can dynamically modify and refine decision making processes based on evolving medical data. The TNM classification is also incorporated, allowing the system to determine tumor classification and recommend appropriate therapies.

For treatment planning, once a diagnosis is confirmed, the system automatically assigns recommendations for therapy. Based on TNM parameters, the logic determines whether the patient should undergo: radiotherapy, chemotherapy, surgery, or palliative care. This automated decision making reduces delays in initiating critical treatment and provides clinicians with immediate guidance.

Ultimately, the system minimizes human error and enhances the accuracy of medical decision making. Through automated rule execution, patient data remains continuously updated, ensuring that doctors can track progression, adapt treatment strategies, and improve patient outcomes efficiently.

```
rule "Change flags for genetic test positive changes"
    no-loop true
when
    $process: ExaminationProcess(geneticTestDone == false, $patient: patient, $examinations: examinations)
    $examination : Examination(
        examinationState == ExaminationState.DONE,
        note matches ".*genetic.*"
    ) from $examinations
    $examinationType : ExaminationType() from $examination.getExaminationTypes()
    $testResult : TestResult(
        value > 0.0,
        description == "positive for changes"
    ) from $examinationType.getTestResults()
then
    modify($process) {
        setGeneticTestDone(true),
        setGeneticTestBadResult(true)
    };
end
```

Fig. 1. Rules for automatically changing flags

5.2 Diagnosis and therapy

The set of rules for automating tumor classification and therapy selection based on the TNM classification system. By analyzing the severity and progression of the disease, the rule "Determine diagnosis based on TNM classification" ensures that patients receive the most appropriate treatment, whether it be surgery, chemotherapy, or radiotherapy. The classification mentioned reduces manual tasks, and ensures accurate data management, allowing healthcare professionals to focus more on patient care.

Similarly, the rule "Determine diagnosis based on examination results" leverages diagnostic findings from CT scans, X-rays, biopsies, and functional lung tests to classify the tumor type and confirm the presence of malignancy (Fig. 2). If any of the ctPositive, rtgPositive, biopsyPositive, or other examination flags indicate concerning results, the system assigns a diagnosis automatically. Based on the nature of the find-ings, the rule determines whether the tumor is intrathoracic pulmonary, intrathoracic non-pulmonary, extrathoracic pulmonary, or extrathoracic non-pulmonary.

Whenever patient data is updated, the rules automatically adjust tumor classification and treatment recommendations. This eliminates the risk of delayed interventions and allows medical professionals to act on the most accurate and up-to-date patient information. Additionally, the automated assignment of tumor type and severity optimizes treatment pathways and ensures that the prescribed therapies align with best clinical practices.

Through this rule-based system, the process of diagnosing and treating lung cancer becomes more structured and efficient cause doctors always can expand the current knowledge base. The automation reduces manual workload for healthcare professionals, allowing them to focus on critical decision making and patient care, ultimately leading to better patient outcomes.

```
rule "Determine diagnosis based on examination results"
    no-loop true
when
    $process: ExaminationProcess(
        ctPositive == true || rtgPositive == true || biopsyPositive == true || spirometryBadResult == true ||
        spiroergometryBadResult == true || dlcoBadResult == true || geneticTestBadResult == true,
        diagnosis == null
    )
    $patient: Patient()
then
    Diagnosis diagnosis = new Diagnosis(null, true, TumorType.INTRATHORACAL_PULMONARY,
    new TNMKlassification(null, 2.0, 1.0, 0.0, LocalDateTime.now()));
    diagnosis.setTumorDetected(true);
    if ($process.isGeneticTestBadResult()) {
        diagnosis.setTumorType(TumorType.INTRATHORACAL_PULMONARY);
    } else if ($process.isDlcoBadResult() || $process.isSpiroergometryBadResult() || $process.isSpirometryBadResult())
    {
        diagnosis.setTumorType(TumorType.INTRATHORACAL_NONPULMONARY);
    } else if ($process.isBiopsyPositive()) {
        diagnosis.setTumorType(TumorType.EXTRATORACAL_PULMONARY);
    } else {
        diagnosis.setTumorType(TumorType.EXTRATORACAL_NONPULMONARY);
    }
    modify($process) {
        setDiagnosis(diagnosis)
    };
    for (Examination exam : $process.getExaminations()) {
        if (exam.getDiagnosis() == null) {
            exam.setDiagnosis(diagnosis);
            System.out.println("Diagnosis assigned for the examination: " + exam.getNote());
        }
    }
end
```

Fig. 2. Rule for diagnosis assignment

5.3 Emergency interventions

```
rule "Generate Temperature Warning"
    no-loop true
when
    $patient : Patient()
    $e1 : TemperatureEvent(patient.getId() == $patient.getId(), temperature > 38.5, $temperature: temperature)
    Number(intValue >= 2) from accumulate(
        $event : TemperatureEvent(
            temperature > 38.5,
            patient.getId() == $patient.getId()
        ) over window:time(12h),
        count($event)
    )
    not(
        TemperatureAlarm(
            patient.getId() == $patient.getId(),
            temperature == $temperature
        )
    )
then
    TemperatureAlarm tempAlarm = new TemperatureAlarm($patient, $temperature, new java.util.Date());
    insert(tempAlarm);
    result.addFact("Temperature Warning Generated for patient: " + $patient.getIme());
    System.out.println("A high temperature warning has been generated for the patient: " + $patient.getIme());
end
```

Fig. 3. Rule for high temperature warning

If the system receives real-time temperature and heart rate data via its dedicated endpoints, it can detect emergency situations. In emergencies, the system uses Complex Event Processing (CEP) rules to detect critical symptoms, such as high body temperature and rapid pulse (Fig. 3). The temperature warning rule is triggered when a patient's temperature exceeds 38.5°C twice within 12 hours, generating an alarm to notify medical staff. Similarly, the pulse warning rule activates an alert when a patient's pulse exceeds 100 BPM twice within the same time frame.

If both critical symptoms persist, the system initiates an emergency intervention, automatically canceling planned therapies, including scheduled surgeries, to prevent complications. The rule **"**Generate Emergency Intervention**"** **(**Fig. 4**)** ensures that medical staff receive an immediate notification, allowing them to take urgent action.

By integrating real-time monitoring and automated response mechanisms, this rule-based system improves patient safety, minimizes response time in emergencies, and ensures that doctors receive timely alerts for critical conditions.

```
rule "Generate Emergency Intervention"
    no-loop true
when
    $pulseAlarm : PulseAlarm($patient: patient)
    $examination : Examination(
        therapy != null,
        therapy.therapyState == TherapyState.PLANNED,
        therapy.therapyType == TherapyType.OPERATION
    ) from $patient.getExaminations()
then
    if($examination.getTherapy()!=null){
        modify($examination.getTherapy()) {
            setTherapyState(TherapyState.CANCELED);
        }
    }
    result.addFact("Emergency Intervention Required for patient: " + $patient.getIme());
    System.out.println("Emergancy for : " + $patient.getIme() + " due to high pulse. The surgery has been canceled. ");
    Alarm alarm = new Alarm(null, $patient, "Emergency Intervention Required", LocalDateTime.now());
    webSocketService.sendAlarm(alarm);
    insert(alarm);
    alarmRepository.save(alarm);
end
```

Fig. 4. Emergency for critical state

6 Conclusion

This paper demonstrates the feasibility of using rule-based systems for the development of an expert system aimed at diagnosing and planning therapy for lung cancer. The system has the potential to significantly improve healthcare quality by providing precise and individualized patient diagnostics. By leveraging rules defined in the Drools engine, the system enables diagnosis, considering factors such as TNM classification and medical test results. It empowers physicians to make informed decisions based on up-to-date clinical data, thereby improving treatment outcomes and reducing the risk of errors in patient assessment. As a rule management mechanism, Drools plays a crucial role in the system's effectiveness, offering flexibility in implementing complex rules and adaptability to changes in medical protocols and guidelines. Integration with a Spring Boot server application and an intuitive user interface provides medical staff with quick and easy access to information, significantly improving workflow efficiency.

Future development of the system could focus on expanding the rule base to include additional diagnostic and therapeutic approaches, as well as enhancing algorithms to handle more complex treatment scenarios. Incorporating machine learning techniques could further improve the system's ability to analyze large datasets and identify patterns that may not be evident through rule-based methods alone. Additionally, integrating the system with other clinical decision support systems and electronic health records could broaden its applicability to a wider range of diseases, including other types of cancer. To streamline data management, the system could be enhanced to automatically update patient data in real-time through seamless integration with healthcare systems, ensuring that all relevant information is synchronized and up-to-date across platforms. Collaboration with medical institutions to validate the system in real-world clinical settings would also be a critical step toward ensuring its reliability and effectiveness. Finally, exploring the potential for mobile or cloud-based deployment could make the system more accessible to healthcare providers in diverse settings, further advancing its impact on global healthcare.

References

[1] Siegel, R. L., Miller, K. D., & Jemal, A. (2020). Cancer statistics, 2020. *CA: A Cancer Journal for Clinicians*, 70(1), 7-30

[2] Hanna, N. H., & Einhorn, L. H. (2020). Personalized treatment for lung cancer: A paradigm shift. *The Lancet Oncology*, 21(9), 1190-1199.

[3] International Association for the Study of Lung Cancer (IASLC). (2016). *The IASLC lung cancer staging project: Proposals for the revision of the TNM stage groupings in the forthcoming (eighth) edition of the TNM classification of lung cancer*. *Journal of Thoracic Oncology, 11*(1), 39-51.

[4] Musen, M. A., Middleton, B., & Greenes, R. A. (2014). Clinical decision-support systems. In *Biomedical Informatics* (pp. 643-674)

[5] Feigenbaum, E. A. (1977). *The art of artificial intelligence: Themes and case studies of knowledge engineering*. Proceedings of the Fifth International Joint Conference on Artificial Intelligence (IJCAI)Detterbeck, F. C., Boffa, D. J., Kim, A. W., & Tanoue, L. T. (2017). The eighth edition lung cancer stage classification. *Chest*, 151(1), 193-203.

[6] Proctor, M., & Frandsen, E. (2016). *Drools: A Rule-Based Approach to Decision Automation and Knowledge Representation*. Packt Publishing.

[7] Abu-naser, S. S., & AlDahdooh, R. M. (2016). Lower back pain expert system diagnosis and treatment. *Journal of Multidisciplinary Engineering Science Studies (JMESS)*, 2(4), 441-446.

[8] Alnattah, A., Jajroudi, M., & Eslami, S. (2024). Exploring rules engines as an artificial intelligence application in medicine: A scoping review. *Unpublished manuscript*

[9] Trajković, A., Ivanović, N., & Nikolić, S. (2022). Creation of dosage for induction in anesthesia using rule-based system. In Zdravković, M.,

Trajanović, M., & Konjović, Z. (Eds.), *ICIST 2022 Proceedings* (pp. 139-144).

[10] Outcomes4Me Free App Created To Better Navigate Cancer Treatment Information. (2022). Forbes

[11] Cancer.Net: Information for Patients and Caregivers - Useful Tools for Providers**.** (2018)

[12] https://github.com/branislavstojkovic70/SBNZ