Information, Computing and Intelligent Systems, 2026, No. 8, 134 – 157
UDC 004.032.26:519.237.8 https://doi.org/10.20535/2786-8729.8.2026.358032

SOFTWARE TECHNOLOGY FOR CLUSTERING STATES BY FEATURE SIMILARITY BASED ON SELF-ORGANIZING KOHONEN MAPS

Oleksii Bychkov
https://orcid.org/0000-0002-9378-9535

Maksym Melnyk*
https://orcid.org/0009-0000-1180-9487

Kateryna Merkulova
https://orcid.org/0000-0001-6347-5191

Volodymyr Petrivskyi
https://orcid.org/0000-0001-9298-8244

Taras Shevchenko National University of Kyiv
Kyiv, Ukraine

*Corresponding author: melnyk.maksym@knu.ua

Received: 19 Apr 2026 / Accepted: 11 May 2026 / Published: 28 May 2026

This paper presents a software technology for clustering high-dimensional states by feature similarity, based on Kohonen self-organizing maps with L2 normalization of binary feature vectors. The technology is realized by autors as Dr.Case program system, a layered software system for automated differential medical diagnosis. The study provides a theoretical foundation for the L2 normalization step in the form of two theorems. The first identifies a systematic bias of the unnormalized Euclidean metric toward the cardinality of binary profiles. The second shows that L2 normalization removes this bias and reduces the pairwise Euclidean distance between binary inputs to a function of structural (cosine) similarity alone. On a database of 844 diseases and 460 symptoms, L2 normalization reduces the self-organizing map quantization error from 2.79 to 0.82. These Quantization Error values measure distances in different geometries and are not directly comparable as absolute distances. Normalization also reduces the topographic error from 0.28 to 0.13 and increases the map fill ratio from 37% to 79%. The software system combines self-organizing map clustering with a candidate selector and a two-branch disease-ranking neural network trained with Focal Loss and Label Smoothing. These components are integrated by an iterative diagnostic cycle with Expected Information Gain question selection, specificity-aware Bayesian answer processing, and rule-based reinforcement for highly specific disease features. The implementation is organized in 16 Python modules with a REST API and a web user interface. The self-organizing map index together with the candidate selector covers 99.5% of the 844 disease catalogue under self-projection (840 of 844 diseases). On a small held-out demonstration set of six clinical cases, the end-to-end system reaches 83.3% Top-1 accuracy.

Keywords: clinical decision support systems, differential diagnosis, feature similarity, Kohonen self-organizing maps, L2 normalization, software architecture.

1. Introduction

Modern medicine has accumulated a very large body of knowledge about diseases and their clinical manifestations. According to the Eleventh Revision of the International Classification of Diseases (ICD-11), more than 55,000 diagnostic entities and classification codes exist. Many of them correspond to conditions with complex clinical manifestations. For a practicing physician, keeping this volume of information in active memory is an extremely demanding task. This load inevitably creates conditions for diagnostic errors, particularly in cases of rare diseases or atypical clinical presentations.

Traditional approaches to systematizing medical knowledge use etiological, pathogenetic, or anatomical principles of classification. Diseases are grouped by cause of occurrence, mechanism of development, or affected organ. From the point of view of differential diagnosis, however, this

approach has substantial limitations. In clinical practice, a physician does not meet the etiology of a disease directly but rather meets its clinical manifestations, namely the symptoms reported by the patient. Two diseases of completely different nature can produce almost identical clinical pictures, which makes them competitors in the process of establishing a diagnosis.

This practical need motivates an alternative organization of medical knowledge: clustering of states (diseases) by the similarity of their feature (symptom) profiles. Under such an approach, influenza and COVID-19, despite different etiologies, appear close to one another because they share clinical manifestations. This matches the real logic of differential diagnosis, in which the physician first forms a set of suspected diagnoses based on symptoms and then narrows that set using additional examinations.

The broader problem that this paper addresses is the clustering of high-dimensional categorical or binary states by feature similarity. Differential medical diagnosis is a representative instance of this problem. The same technology applies to other domains in which a state is characterized by the presence or absence of a set of features. Examples include fault diagnosis in engineering systems and document classification by topic signatures. The aim of this research is to develop a complete software technology for such clustering and to demonstrate it on the medical diagnosis task. Particular attention is paid to the choice of metric and the normalization of data, which proved decisive for obtaining semantically coherent clustering results. Attention is also paid to the software architecture that integrates unsupervised clustering, supervised ranking, and an iterative dialogue with the user.

Over the past decade, machine learning and artificial intelligence have achieved strong results in the medical field. Deep neural networks produce near-human accuracy on image-based pathology recognition, natural language processing systems analyze electronic health records, and predictive algorithms estimate the risk of disease development. There is, however, a gap between academic results and the real needs of clinical practice.

Ordinary citizens increasingly turn to the Internet with questions about their health. They look for an accessible and reliable tool for a preliminary assessment of symptoms, one that could indicate whether urgent medical attention is needed or whether the condition can wait. Many publicly available symptom checkers still provide limited transparency and may fail to capture complex relationships among symptoms.

Primary-care physicians must navigate thousands of possible diagnoses every day. Rare diseases present a particularly difficult situation, because the physician may simply not remember their existence. An intelligent assistant that, from a given set of symptoms, suggests a list of probable diagnoses, including rare ones, could improve diagnostic quality and reduce missed conditions.

Medical students need effective simulators for developing clinical reasoning. Traditional teaching from textbooks and lectures does not by itself provide enough practical experience with differential diagnosis. An interactive system that models the diagnostic process through successive symptom clarification could be used as a training tool.

These considerations together justify the development of an intelligent medical diagnostic system based on methods of clustering diseases by symptom similarity that produce interpretable results. At the same time, such a system must be packaged as a complete software technology. This technology requires a clear software architecture, stable interfaces, and a set of independent modules that can be reused, replaced, or extended. These considerations make the problem addressed in this paper a timely one. The body of formalized medical knowledge continues to expand, while the working memory of a physician does not, so the rate of diagnostic error stays high and falls disproportionately on rare diseases and atypical presentations. Classification schemes that organize this knowledge by etiology, pathogenesis, or anatomy do not follow the symptom-based reasoning along which differential diagnosis proceeds. A method that orders diseases by the similarity of their symptom profiles answers a demand that present-day diagnostic tools satisfy only in part. Such a method is delivered as a software technology with an interpretable two-dimensional representation and stable module interfaces. The

task is at once practically motivated and methodologically open, which is what the present study sets out to address.

2. Literature review and problem statement

Kohonen self-organizing maps, introduced by [1], have been applied widely in medical informatics. In [2] later described how clustering can be derived from a trained Self Organising MAP (SOM), giving on which the present study builds. A closer reading of existing studies, however, exposes limitations that call for new approaches.

In [3] proposed a combined SOM+Fuzzy Support Vector Machine (Fuzzy SVM) pipeline for diagnosing coronary heart disease with incremental updates. The authors used Principal Component Analysis (PCA) for dimensionality reduction, followed by SOM clustering and Fuzzy SVM classification. Experiments on the Cleveland and Statlog datasets gave a clear accuracy gain over baseline methods. For the goals of this study, the work has two limitations. First, it targets a single disease (coronary heart disease), whereas the present study aims at a general-purpose system for hundreds of diagnoses. Second, the authors do not study cluster interpretation or its use for differential diagnosis.

In [4] developed the SOM Net decision support system for planning psychiatric care at the regional level. SOM was used there to visualize and analyze relationships among structural, process, and outcome indicators of mental-health systems. The system was useful for administrative decisions, yet it operates at the population level, not at the level of individual diagnosis. Its inputs are aggregate statistical indicators, not symptoms of particular diseases.

In [5] applied SOM to cluster analysis in a multi-disease diagnosis task. The authors showed that SOM has advantages over traditional clustering methods such as k–means, thanks to its ability to preserve topological structure and to work with incomplete information. This study is the closest to the present problem. The authors, however, do not examine the question of binary symptom-vector normalization, which this study later shows to be decisive in its Euclidean SOM configuration for obtaining semantically coherent clusters.

In [6] used SOM to analyze clinical data on osteoporosis. The method revealed hidden patterns in densitometry data and identified groups of patients with elevated fracture risk. The study is limited to one nosology and uses quantitative densitometry indicators rather than binary symptoms.

In [7] applied SOM to the identification of mental disorders, showing the method's usefulness for working with unstructured clinical data. The scope of the work is limited to psychiatric diagnoses, and the problem of scaling to the full range of medical conditions is not addressed.

Among more recent applications of SOM in medical knowledge representation, the authors of article [8] trained a self-organizing map on the full set of Medical Subject Headings (MeSH) annotations of peer-reviewed articles in Medline. They then validated the resulting MedSOM by projecting the reference lists from ten editions of a core psychiatric textbook. The work confirms that SOM can produce meaningful 2D representations of medical knowledge at large scale. The input there, however, is bibliographic metadata rather than clinical symptom vectors, and the goal is curriculum validation rather than patient diagnosis.

Modern automated differential-diagnosis systems have shifted toward deep learning models. In [9] developed a clinical decision support system based on learning–to–rank using TensorFlow Ranking with Approximate Normalized Discounted Cumulative Gain (NDCG) loss. The system takes as input a physician-entered list of symptoms, findings, and test results and returns a ranked list of possible diseases. The authors report gains over conventional baselines and discuss interpretability at a high level, but the architecture does not include of the kind proposed in this study.

In [10] compared Decision Tree, Random Forest, Naive Bayes, Logistic Regression, and K–Nearest Neighbors classifiers on a symptom-based health-checker task, using clinical vignettes for external validation. The reported accuracy is high for the evaluated disease subset. The underlying dataset,

however, covers only 10 diseases with 9,572 samples, an order of magnitude smaller than the catalogue of 844 diseases targeted in this study.

The authors of paper [11] proposed the Hypergraph Clustering with multi-classification Label Entropy for Multi-Label Classification (HCLE-MLC) method. It combines hypergraph clustering with multi-classification label entropy for multi-label disease diagnosis in the context of Traditional Chinese Medicine syndrome (TCM syndromes) differentiation. The method constructs a symptom hypergraph and applies a clustering-optimized hypergraph attention network. Their work confirms the value of combining clustering with multi-label ranking, but the domain (TCM syndromes) and the clustering technique (hypergraphs) differ from those of the present approach.

Several studies of commercial Artificial Intellect (AI) symptom checkers have appeared recently. The authors of work [12] performed a three-year longitudinal assessment of differential-diagnosis lists produced by an AI-based symptom checker in outpatient practice. They reported no improvement in diagnostic accuracy over time and identified independent associations of lower accuracy with rare diseases and atypical presentations. The authors of paper [13] carried out a multicenter randomized controlled trial comparing the Ada mobile symptom checker and the Rheport web tool in rheumatology. They reported heterogeneous diagnostic accuracy for individual diagnoses and poor inter-tool agreement on the presence of any inflammatory rheumatic disease. The authors of paper [14] compared Ada and WebMD symptom checkers, ChatGPT 3.5, and ChatGPT 4.0 against emergency-department physician diagnoses. They found that dedicated symptom checkers outperformed general-purpose chatbots on diagnostic accuracy, while ChatGPT-4 showed higher triage safety. These studies together show that the quality of existing systems is still uneven, particularly for rare diseases, and that a principled approach to candidate selection and interpretation is worth pursuing.

The authors of work [15] compared ChatGPT with GPT-3.5 and GPT-4 against primary-treating resident physicians in an internal-medicine emergency department. They reported that GPT-4 outperformed GPT-3.5 and, on specific case subsets, matched or exceeded the accuracy of resident physicians. The strength of this line of work is the use of real admissions. Its weakness for the present study is that large language models do not expose an interpretable intermediate representation of the clinical space.

The authors of [16] reviewed the use of machine learning for rare diseases over the decade ending in 2022. The review finds that predictive modeling with deep learning has been applied to a growing number of rare conditions. Data scarcity and the heterogeneity of clinical presentations, however, remain open problems. Rare diseases are exactly the category that general-purpose symptom checkers tend to miss. This gap reinforces the motivation for a method that clusters diseases by symptom structure rather than by class frequency alone.

A review in [17] of clinical decision-support systems based on explainable artificial intelligence (XAI) discusses requirements for interpretability in medical applications. The authors of article [18] performed a separate systematic review of XAI in healthcare over the period 2011–2022. The review documented the uptake of post-hoc explanation methods such as SHapley Additive exPlanations (SHAP), Local Interpretable Model-agnostic Explanation (LIME), and Gradient-weighted Class Activation Mapping (Grad-CAM). It also noted persistent gaps in clinician-centered evaluation. SOM is relevant in this context because it provides visualization and interpretation through its two-dimensional map, in contrast to the black-box behavior of deep learning models.

Focal Loss [19] and Label Smoothing [20]], introduced originally in the computer-vision literature, are widely adopted regularization techniques for training deep classifiers on imbalanced tasks.

The literature review above shows that existing applications of SOM in medicine share three common limitations:

– a focus on narrow subject areas, namely a single disease or a single class of diseases;
– the use of quantitative indicators in place of binary symptoms;

– an absence of theoretical justification for the choice of metric and normalization when the data are sparse and binary.

Modern symptom checkers and large language models achieve useful accuracy in parts of the domain but give limited interpretability and often underperform on rare diseases. None of the reviewed works offers an integrated software technology for clustering hundreds of states by feature similarity with subsequent use in iterative differential analysis. The present study addresses this gap.

3. The aim and objectives of the study

The aim of the study is to provide a simplified process of high-quality diagnostics for automated differential medical diagnosis. The technology should raise the quality of disease clustering, the accuracy of candidate-diagnosis ranking, and the interpretability of diagnostic results relative to existing symptom-based systems.

To achieve this aim, the following tasks are addressed:

– to justify theoretically the choice of distance metric and the normalization of binary symptom vectors, establishing the conditions under which Euclidean distance reflects structural feature overlap rather than the cardinality of a symptom profile;

– to build a clustering model of diseases by symptom similarity on a Kohonen self-organizing map with normalized binary feature vectors;

– to develop a candidate-selection mechanism and a two-branch ranking neural network with Focal Loss and Label Smoothing for ordering probable diagnoses;

– to design an iterative diagnostic cycle with Expected Information Gain question selection, specificity-aware Bayesian answer processing, and rule-based reinforcement for pathognomonic features;

– to implement the technology as a layered software system of loosely coupled modules with a representational-state-transfer service interface and a web user interface;

– to confirm, on the disease–symptom database of 844 diseases, the improvement in clustering quality from normalization using quantization error, topographic error, and fill ratio, and to validate the semantic coherence of the resulting disease clusters as the basis of the system's interpretability;

– to demonstrate the end-to-end behaviour of the integrated diagnostic system on a held-out set of clinical cases using Top-1 and Top-3 accuracy, mean reciprocal rank, and the average number of clarifying questions, and to measure the candidate self-projection recall of the self-organizing-map index and selector over the full disease catalogue.

4. Materials and methods of the study on disease clustering by symptom similarity for differential diagnosis

4.1. Object, subject, and hypothesis of the study

The object of the study is the process of automated differential medical diagnosis. In this process, diseases are organized by the similarity of their symptom profiles rather than by etiology, pathogenesis, or affected organ.

The subject of the study is the set of models, methods, and the software technology that realize this organization. The core is the clustering of diseases on a Kohonen self-organizing map over normalized binary symptom vectors. To this are added the candidate-selection and ranking mechanisms that turn the resulting structure into a ranked list of probable diagnoses.

The working hypothesis is that L2 normalization of binary symptom vectors before training removes the dependence of the Euclidean metric on the number of symptoms in a profile. As a result, the clusters group diseases by the structure of their symptom profiles rather than by symptom count. Under this hypothesis, normalization raises clustering quality and yields semantically coherent disease clusters that serve as a sound basis for differential diagnosis. The hypothesis is examined theoretically in Section 4.2 (Theorems 1 and 2) and tested empirically in Section 5 (Tables 6 and 7).

4.2. Materials and methods of the study

The study combines methods of unsupervised and supervised machine learning within a single software system. The central element of the proposed approach is the Kohonen self-organizing map, which is used to build a two-dimensional representation of the high-dimensional symptom space.

A self-organizing map is a neural network composed of a two-dimensional lattice of neurons, each of which holds a weight vector of the same dimensionality as the input. During training, the self-organizing map performs competitive learning. For each input vector, it identifies the nearest neuron (the Best Matching Unit, BMU) and then updates the weights of that BMU and its topological neighbors. The trained map preserves the topological structure of the input data, so that similar objects project onto nearby regions of the map.

For ranking candidate diagnoses, the study uses a multilayer neural network with a two-branch architecture, denoted TwoBranchNN_BMU_Focal. A distinguishing feature of this architecture is the presence of two parallel processing branches. The first branch receives the patient's binary symptom vector, and the second receives the normalized coordinates of the nearest self-organizing map unit. This arrangement allows the neural network to use both direct symptom information and contextual information about cluster membership of the case.

Clustering quality is evaluated with three standard self-organizing map metrics :

- quantization error (QE), the average distance from objects to their Best Matching Units (BMUs);
- topographic error (TE), a measure of how well the topological structure is preserved;
- the map fill ratio (Fill), the fraction of activated units, defined in (1).

$$\text{Fill} = \frac{\left|\{u \in U : \text{diseases}(u) \neq \varnothing\}\right|}{|U|}. \tag{1}$$

In (1), U is the set of all self-organizing map units, and diseases (u) is the set of diseases whose BMU is u. Diagnostic quality is evaluated with Top–k Accuracy, Mean Reciprocal Rank (MRR), and candidate recall.

Each disease d in the database is represented by a binary symptom vector $x_d \in \{0,1\}^{460}$. Each component equals 1 if the corresponding symptom is typical for that disease and 0 otherwise. When a SOM is trained on such vectors with the standard Euclidean metric, a fundamental problem arises that can be illustrated with the following example.

Consider four diseases with different numbers of symptoms. A schematic example is shown in Table 1. The cluster labels (A, B, C) are illustrative placeholders that summarize the qualitative SOM behavior expected on unnormalized binary data; they are not the output of a specific SOM run.

Table 1. Illustrative norms of symptom vectors without L2 normalization (schematic)

Disease	**Number of symptoms**	$\|x_d\|_2$	**SOM cluster**
Influenza	3	$\sqrt{3} \approx 1.73$	Cluster A
Common cold	2	$\sqrt{2} \approx 1.41$	Cluster B
Pneumonia	10	$\sqrt{10} \approx 3.16$	Cluster C
Diabetes mellitus	10	$\sqrt{10} \approx 3.16$	Cluster C (!)

As the Table 1 shows, pneumonia and diabetes mellitus have the same vector norm ($\sqrt{10} \approx 3.16$), although their symptoms are completely different. Pneumonia is a respiratory disease with symptoms such as fever, cough, and shortness of breath. Diabetes mellitus, in contrast, is an endocrine disease with symptoms such as increased thirst, frequent urination, and fatigue. Equal norms alone do not

make these two vectors close in Euclidean distance: for disjoint symptom sets the squared distance is $\|x-y\|_2^2 = k_x + k_y - 2c = 10+10-0 = 20$. The clustering bias is therefore caused not by pairwise vector proximity but by the geometry of SOM prototypes. When a unit's weights converge to approximately uniform average values, the distances from the unit to both vectors become similar despite the absence of shared symptoms. The two diseases can then be assigned to neighboring units that are dominated by the cardinality component of the metric rather than by symptom overlap.

A critical observation is that during SOM training the unit weights converge toward the average values of the input data. Suppose a unit's weights are approximately uniform, for example near 0.5 on all components. The distance from this unit to pneumonia and to diabetes mellitus is then approximately equal, despite the complete absence of shared symptoms. Without normalization, Euclidean distances are affected by both symptom overlap and vector cardinality. This can distort SOM organization and cause diseases with similar profile sizes to compete for similar regions of the map, especially when prototype weights represent averaged patterns.

Notation. Let $x, y \in \{0,1\}^m$ be nonzero binary feature vectors. We write $S_x = \{i : x_i = 1\}$ and $S_y = \{i : y_i = 1\}$ for the sets of active features, and introduce

$$k_x = |S_x|, k_y = \left|S_y\right|, c = |S_x \cap S_y|.$$

The quantity k_x is the number of active features in x (and coincides with both $\|x\|_0$ and $\|x\|_2^2$ on binary data); c is the number of shared active features. The structural similarity coefficient of the two profiles is defined in (2)

$$\rho(x,y) = \frac{c}{\sqrt{k_x k_y}}. \tag{2}$$

It is the cosine similarity of the vectors and, for binary data, coincides with the Ochiai similarity of the sets S_x and S_y. The motivating example above suggests that, on raw binary data, Euclidean distance mixes two distinct effects. These are the overlap of the feature sets (captured by c) and the absolute number of active features (captured by k_x, k_y). The first theorem below makes this mixing precise. It shows that the resulting bias is a property of the metric itself, not a peculiarity of a particular dataset.

Purpose of Theorem 1. The theorem is a negative result: its role is to identify, in closed form, the mechanism by which the raw Euclidean metric on binary data departs from pure structural comparison. It shows that unnormalized Euclidean distance is sensitive to the cardinalities k_x, k_y even when the shared-overlap ratio ρ is held fixed. As a consequence, two pairs of profiles with identical structural similarity produce different distances whenever they differ in scale. In the context of SOM clustering, this is the algebraic source of the clustering bias toward "profile size." This bias is illustrated schematically in Table 1 and observed empirically on the full 844-disease database. The theorem therefore justifies the claim that the observed distortion is not accidental but is built into the chosen metric. It also motivates the corrective step.

Theorem 1 (Bias of unnormalized Euclidean distance toward profile cardinality). For any nonzero binary vectors $x, y \in \{0,1\}^m$

$$\|x-y\|_2^2 = k_x + k_y - 2c. \tag{3}$$

In particular:

1. for fixed overlap c, the distance $\|x-y\|_2^2$ is strictly increasing in the sum $k_x + k_y$;
2. for fixed structural similarity $\rho(x,y) = \rho_0$, the unnormalized Euclidean distance is in general not invariant to k_x, k_y: pairs with the same ρ_0 but different cardinalities produce different distances.

In the symmetric case $k_x = k_y = n$

$$\|x-y\|_2^2 = 2n\big(1-\rho(x,y)\big). \tag{4}$$

Proof. Expand the squared Euclidean distance via the standard identity (5), (6)

$$\|x - y\|_2^2 = \|x\|_2^2 + \|y\|_2^2 - 2\langle x, y\rangle. \tag{5}$$

Since $x_i, y_i \in \{0, 1\}$, the relations $x_i^2 = x_i$ and $y_i^2 = y_i$ hold componentwise. Therefore

$$\|x\|_2^2 = \sum_{i=1}^{m} x_i^2 = \sum_{i=1}^{m} x_i = |S_x| = k_x, \quad \|y\|_2^2 = k_y. \tag{6}$$

For the inner product, $x_i y_i = 1$ if and only if $x_i = 1$ and $y_i = 1$, that is, $i \in S_x \cap S_y$; otherwise $x_i y_i = 0$. Hence (7)

$$\langle x, y\rangle = \sum_{i=1}^{m} x_i y_i = |S_x \cap S_y| = c. \tag{7}$$

Substituting (4)) and (5) into (3) yields (8)

$$\|x - y\|_2^2 = k_x + k_y - 2c, \tag{8}$$

which proves (1).

For claim 1, fix c. Then the right-hand side of (1) is the affine function $\big(k_x + k_y\big) - 2c$ of the sum $k_x + k_y$, with positive slope 1 in $k_x + k_y$, and is therefore strictly increasing in that sum.

For claim 2 and (2), assume $k_x = k_y = n$. Then $\sqrt{k_x k_y} = n$ and, by definition of ρ

$$c = \rho(x, y) \cdot \sqrt{k_x k_y} = \rho(x, y)n. \tag{9}$$

Substituting (9) into (1),

$$\|x - y\|_2^2 = n + n - 2\rho(x, y)n = 2n\big(1 - \rho(x, y)\big), \tag{10}$$

which proves (2). To see that the distance is not ρ-invariant across scales, take two pairs (x, y) and (u, v) with $\rho\,(x, y) = \rho\,(u, v) = \rho_0$, $k_x = k_y = n$, and $k_u = k_v = m$. By (2),

$$\|x - y\|_2^2 = 2n\big(1 - \rho_0\big), \quad \|u - v\|_2^2 = 2m\big(1 - \rho_0\big) \tag{11}$$

and the two distances (11) coincide only when $n = m$.

Interpretation. Theorem 1 states that the unnormalized squared Euclidean distance between two binary profiles is a sum of a structural term $(-2c)$ and a cardinality term $(k_x + k_y)$. Even profiles that are structurally identical in the ρ–sense produce different distances whenever they live at different scales. The distortion illustrated by Table 1 is therefore inherent in the metric.

To remove the magnitude term identified in Theorem 1, apply L2 normalization to each feature vector before SOM training

$$\hat{x} = \frac{x}{\|x\|_2}, \quad \hat{y} = \frac{y}{\|y\|_2}. \tag{12}$$

All normalized vectors have unit norm and lie on the surface of the unit sphere in $\mathbb{R}^m$. Table 2 continues the schematic example of Table 1. The norms of the normalized vectors are all equal to one, and in the schematic the four diseases now group by symptom structure rather than by symptom count. The actual SOM behavior on the full 844 disease database.

After L2 normalization, all vectors have the same unit norm, independent of the number of symptoms. Influenza, common cold, and pneumonia share respiratory symptoms and fall into a single “Respiratory” cluster. Diabetes mellitus, with a completely different symptom set, is correctly

Table 2. Illustrative norms of symptom vectors with L2 normalization (schematic)

Disease	Number of symptoms	$\|\hat{x}_d\|_2$	SOM cluster
Influenza	3	1.00	Respiratory
Common cold	2	1.00	Respiratory
Pneumonia	10	1.00	Respiratory
Diabetes mellitus	10	1.00	Endocrine

placed in a separate "Endocrine" cluster. This matches the clinical logic of differential diagnosis. The next theorem explains algebraically why this happens.

Purpose of Theorem 2. The theorem is a positive result, complementary to Theorem 1: it establishes that L2 normalization eliminates exactly the magnitude term that caused the bias. After normalization, the squared Euclidean distance becomes a function of $\rho(x, y)$ alone, so pairs of profiles with identical structural similarity produce identical distances regardless of their cardinalities. The theorem thus provides the mathematical justification for using L2 normalization as a preprocessing step before SOM training on binary feature vectors. It also establishes the equivalence between Euclidean comparison on the unit sphere and cosine similarity on the original space.

Theorem 2 (Structural invariance of Euclidean distance after L2 normalization). For any nonzero binary vectors $x, y \in \{0,1\}^m$, the squared Euclidean distance between their L2–normalized versions satisfies

$$\|\hat{x} - \hat{y}\|_2^2 = 2\big(1 - \rho(x, y)\big) = 2\left(1 - \frac{c}{\sqrt{k_x k_y}}\right). \tag{13}$$

In particular, $\|\hat{x} - \hat{y}\|_2^2$ depends on the pair (x, y) only through the structural similarity $\rho(x, y)$, and not on k_x or k_y separately. Consequently, if two pairs of profiles satisfy $\rho(x, y) = \rho(u, v)$, then (14)

$$\|\hat{x} - \hat{y}\|_2^2 = \|\hat{u} - \hat{v}\|_2^2, \tag{14}$$

regardless of the individual cardinalities k_x, k_y, k_u, k_v.

Proof. Apply the expansion (3) to the normalized vectors (15)

$$\|\hat{x} - \hat{y}\|_2^2 = \|\hat{x}\|_2^2 + \|\hat{y}\|_2^2 - 2\langle \hat{x}, \hat{y} \rangle. \tag{15}$$

By definition (6), $\|\hat{x}\|_2 = \|\hat{y}\|_2 = 1$, whence $\|\hat{x}\|_2^2 = \|\hat{y}\|_2^2 = 1$, and (9) reduces to (16)

$$\|\hat{x} - \hat{y}\|_2^2 = 2 - 2\langle \hat{x}, \hat{y} \rangle. \tag{16}$$

The inner product of the normalized vectors is (17)

$$\langle \hat{x}, \hat{y} \rangle = \left\langle \frac{x}{\|x\|_2}, \frac{y}{\|y\|_2} \right\rangle = \frac{\langle x, y \rangle}{\|x\|_2 \|y\|_2}. \tag{17}$$

Using the binarity identities (4) and (5) established in the proof of Theorem 1, $\|x\|_2 = \sqrt{k_x}$, $\|y\|_2 = \sqrt{k_y}$, and $x, y = c$, so (18)

$$\langle \hat{x}, \hat{y} \rangle = \frac{c}{\sqrt{k_x k_y}} = \rho(x, y). \tag{18}$$

Combining (10) with (12) gives (19)

$$\|\hat{x} - \hat{y}\|_2^2 = 2\big(1 - \rho(x, y)\big), \tag{19}$$

which is the first equality in (7); the second equality is the definition of ρ. The invariance claim (8) is immediate: the right-hand side of (7) depends on (x, y) only through $\rho(x, y)$, so two pairs with equal ρ produce equal normalized distances.

Interpretation. Comparing Theorem 1 and Theorem 2 side by side makes the role of normalization explicit. On raw vectors,

$$\|x - y\|_2^2 = k_x + k_y - 2c, \tag{20}$$

so the distance contains both a structural term and a cardinality term. On normalized vectors

$$\|\hat{x} - \hat{y}\|_2^2 = 2\big(1 - \rho\big), \tag{21}$$

so only the structural term remains. L2 normalization is therefore precisely the correction that removes the cardinality bias. It is not an ad hoc preprocessing step but the operation that makes the Euclidean metric agree with the structural similarity that the clustering task actually requires.

Corollary 1. On L2–normalized binary feature vectors, the Euclidean metric induces the same ordering on pairs as cosine similarity: for any two pairs (x, y) and (u, v)

$$\|\hat{x} - \hat{y}\|_2 \leq \|\hat{u} - \hat{v}\|_2 \Leftrightarrow \rho(x, y) \geq \rho(u, v). \tag{22}$$

Corollary 2. For pairwise comparisons between L2-normalized binary input vectors, the Euclidean metric induces the same ordering as cosine/Ochiai similarity. Input-input comparisons therefore depend on structural feature overlap rather than on raw feature count. In SOM training, this normalization changes the geometry of the input space and removes the magnitude term from pairwise input distances. The equivalence to cosine similarity, however, is exact only for input-input comparisons. For input-prototype distances it should be interpreted as an induced geometric effect, since SOM weight vectors w_u are real-valued and not in general of unit norm.

Corollary 3. Euclidean distance on L2–normalized binary vectors is equivalent, up to the monotone transformation $d^2 \mapsto 1 - d^2/2$, to the cosine (Ochiai) similarity of the underlying feature sets. This similarity is the natural choice for sparse high-dimensional binary data.

The SOM is realized as a 15×15 grid of neurons (225 units), each with a 460 dimensional weight vector. Training runs for 1000 epochs with a Gaussian neighborhood function. The learning rate decays exponentially from 0.5 to 0.01, and the neighborhood radius decays exponentially from 7.5 (half of the grid size) to 0.5. Weights are initialized with Principal Component Analysis (PCA), which gives faster convergence than random initialization on sparse binary data. All inputs are L2–normalized prior to training and projection, x denotes an L2–normalized input. At each training step, the Best Matching Unit for an input x is found as (23)

$$\mathrm{BMU}(x) = \arg\min_u \|x - w_u\|^2. \tag{23}$$

Weights of the BMU and its neighbors are updated as (24)

$$w_u(t+1) = w_u(t) + \eta(t) h(u, \mathrm{mathrmBMU}, t)\big(x - w_u(t)\big), \tag{24}$$

where the neighborhood kernel is (25)

$$h(u, \mathrm{BMU}, t) = \exp\left(-\frac{\|r_u - r_{\mathrm{BMU}}\|^2}{2\sigma^2(t)}\right), \tag{25}$$

with $\eta(t) = \eta_0 \exp\left(-t/\tau_\eta\right)$ and $\sigma(t) = \sigma_0 \exp\left(-t/\tau_\sigma\right)$.

After training, the system builds a SOM index that maps each active unit to the set of diseases whose vectors project onto it. The index is the main data structure for downstream candidate selection.

The Candidate Selector takes the patient's projection on the SOM and returns a subset of candidate diseases. Its internal sanity-check metric, the *self-projection recall*, measures the fraction of diseases d for which the disease vector x_d. For such a disease, projected onto the SOM and passed through the

selector, produces a candidate set D_{cand} that contains d itself. On the full database of 844 diseases, this metric reaches 99.5% (840 out of 844 diseases). This establishes that the SOM index plus selector policy cover the disease catalogue almost completely, and that such coverage is a prerequisite for any downstream recall on real patients. The self-projection recall is an upper bound on recall for unseen patient inputs, not a substitute for it.

For each SOM unit u, the distance from the L2–normalized patient vector to the unit weights is $d_u = \|\hat{x}_{\text{patient}} - w_u\|$, and the membership is computed as a softmax over squared distances

$$m_u = \frac{\exp(-d_u^2/\lambda)}{\sum_v \exp(-d_v^2/\lambda)}, \tag{26}$$

where $\lambda = 1.0$ is a sharpness parameter.

The selector applies a combined policy. The units are first ranked by membership. A set S is formed by taking units one by one in order of decreasing membership until the cumulative mass reaches $\alpha = 0.9$ (cumulative-mass policy). If fewer than $k = 6$ units are taken at that point, the selection is extended to the top k units (top–k guarantee). Units with membership below $\tau = 0.01$ are excluded (threshold cutoff). The candidate set D_{cand} is then the union of diseases assigned to units in S. In the 844 disease database, the selector returns on average 35 candidates per case, which is an order of magnitude reduction of the search space.

4.3. Disease-ranking neural network

The ranking component is a two-branch neural network, TwoBranchNN_BMU_Focal, used as a single-label multiclass disease ranker over the 844 disease catalogue. The symptom branch receives the binary vector $x \in \mathbb{R}^{460}$ and passes it through the subnetwork Linear (460, 256) → BatchNorm → ReLU → Dropout (0.3) → Linear (256, 128) → BatchNorm → ReLU → Dropout (0.3).

The BMU branch receives the normalized BMU coordinates $(y/H, x/W) \in \mathbb{R}^2$ and passes them through Linear (2, 32) → BatchNorm → ReLU → Dropout (0.2) → Linear (32, 16) → BatchNorm → ReLU.

The two outputs are concatenated into a 144 dimensional vector, passed through a combined Linear (144, 128) → BatchNorm → ReLU → Dropout (0.3) block, and then into a final Linear (128, 844) layer that produces logits for the 844 diseases. A softmax over the 844 logits is applied externally when probabilities are required; the network is trained for single-label multiclass ranking, not for multilabel binary classification.

Training uses Focal Loss [19] in its multiclass formulation, combined with Label Smoothing [20]. Let $C = 844$ be the number of diseases, $y \in \{1, \ldots, C\}$ the true class, and p_i the softmax probability for class i. The smoothed target distribution is (27)

$$q_i = \begin{cases} 1 - \varepsilon + \frac{\varepsilon}{C}, & i = y, \\ \frac{\varepsilon}{C}, & i \neq y. \end{cases} \tag{27}$$

with $\varepsilon = 0.1$. The focal modulation factor is applied per class to the smoothed cross-entropy (28)

$$\text{FL}(p, y) = -\alpha \sum_{i=1}^{C} q_i (1 - p_i)^{\gamma} \log p_i. \tag{28}$$

with focusing parameter $\gamma = 2.0$ and global scaling factor $\alpha = 0.25$. The same α is applied to every class, so it is a uniform loss scaling and not a class-balancing weight. Class imbalance during ranker training is mitigated, rather than fully corrected. Two mechanisms contribute: the focusing factor $(1 - p_i)^{\gamma}$, which down-weights well-classified examples, together with the candidate filtering performed by the Candidate Selector. A genuinely class-balanced variant would require per-class weights α_i inside the sum or a balanced sampling strategy. For the unsmoothed case ($\varepsilon = 0$), only

the $i = y$ term is non-zero and (17) reduces to the standard multiclass focal loss $-\alpha(1 - p_y)^\gamma \log p_y$. The optimizer is AdamW with learning rate 10^{-3}, weight decay 10^{-4}, and a Cosine Annealing Warm Restarts schedule ($T_0 = 10$, $T_{\text{mult}} = 2$, $\eta_{\min} = 10^{-6}$). Batch size is 64, gradient clipping is 1.0, and training runs for up to 100 epochs with early stopping patience of 10. The data split is 75%/15%/10% for train/validation/test.

The Question Engine decides which symptom to ask the user about next. The base criterion is Expected Information Gain (EIG)

$$H(\hat{y}) = -\sum_i \hat{y}_i \log \hat{y}_i, \tag{29}$$

$$\text{EIG}(q) = H\big(\hat{y}^{(t)}\big) - \sum_a P(a \mid q) H\big(\hat{y}^{(t+1)} \mid a\big), \tag{30}$$

$$q^* = \arg\max_q \text{EIG}(q). \tag{31}$$

The answer probability $P(a \mid q)$ is estimated from the current posterior over candidate diseases and the same likelihood table used by BayesianAnswerProcessor (Table 4) (32)

$$P(a \mid q) = \sum_{d \in D_{\text{cand}}} P\big(a \mid h(d), s(q)\big)\hat{y}_d. \tag{32}$$

Here $h(d) = 1$ if $q \in \text{symptoms}(d)$ and $h(d) = 0$ otherwise. The EIG simulation considers only the answers $a \in \{\text{Yes}, \text{No}\}$. The "Unknown" answer is admissible at the dialogue level but is treated as a non-update (no change to $\hat{y}$) and therefore does not enter the entropy reduction sum. This makes EIG an optimistic two-outcome planning criterion: frequent Unknown answers reduce the realized information gain in practice below the value predicted by (18). The selector simulates the two answers Yes and No for each candidate question, computes the expected posterior entropy, and chooses the symptom with the largest EIG.

The v2.0 component `FocusedQuestionSelector` extends the base criterion with specificity-aware multipliers. Let $s(q) = |\{d \in D : q \in \text{symptoms}(d)\}|$ be the number of diseases for which symptom q is listed. Let $\text{splits}(q)$ be true when the presence of q separates the top–1 from the top–2 candidate (that is, exactly one of them has q). The adjusted score is given in Table 3.

Table 3. Specificity bonuses in `FocusedQuestionSelector`

Condition	Multiplier	Interpretation
Very specific ($s \le 5$) and splits top-2	×10	Rare symptom, ideal for differentiation
Specific ($s \le 20$) and splits top-2	×5	Moderately rare symptom
General ($s > 20$) and splits top-2	×2	Common symptom but discriminative
Does not split top-2	×1	Base EIG without bonus

The rationale is that a rare symptom that separates the two leading hypotheses yields more useful information per question than a common symptom that does not discriminate between them.

Three further v2.0 components improve the handling of user answers.

The `BayesianAnswerProcessor` replaces a symmetric update of the hypothesis probabilities with a specificity-aware one. Let $p(d)$ be the prior probability of disease d before the answer, $h(d) = 1$ if $q \in \text{symptoms}(d)$ and 0 otherwise. The likelihood $P(a \mid h, s(q))$ is tabulated in Table 4. The

posterior is then

$$p(d \mid a) = \frac{P\big(a \mid h(d), s(q)\big)p(d)}{\sum_{d'} P\big(a \mid h(d'), s(q)\big)p(d')}. \tag{33}$$

Table 4. Likelihoods used in the `BayesianAnswerProcessor`

Specificity	$P(\textbf{Yes} \mid h = 1)$	$P(\textbf{Yes} \mid h = 0)$	$P(\textbf{No} \mid h = 1)$	$P(\textbf{No} \mid h = 0)$
Very specific ($s \le 5$)	0.90	0.05	0.10	0.95
Specific ($s \le 20$)	0.80	0.25	0.20	0.75
General ($s > 20$)	0.65	0.50	0.35	0.50

By construction, $P(\text{Yes} \mid h) + P(\text{No} \mid h) = 1$ in every row for each value of $h \in \{0, 1\}$.

The `SymptomDrivenDiscovery` component addresses cases in which the neural ranker assigns a very low score to the correct diagnosis in the first ranking. Without intervention, such a diagnosis would never be reached by standard iteration. When the user confirms a highly specific symptom q (low $s(q)$), the component lifts the probability of each disease in the set $\{d : q \in \text{symptoms}(d)\}$ to a floor value that depends on the symptom's specificity. The floor 30% for $s = 1$, 15% for $s \in [2, 3]$, 8% for $s \in [4, 5]$, 5% for $s \in [6, 10]$. The list is then renormalized. This creates a safety net for rare diseases that would otherwise be hidden by high-frequency competitors.

The `Rule-based Boost` stores a dictionary of 207 highly specific symptoms drawn from the medical literature, the subset with $s = 1$ corresponding to features that are unique within the current database. On each iteration, the component multiplies the probabilities of diseases linked to a confirmed entry of this dictionary by a fixed factor: $\times$ 3.0 when $s = 1$, $\times$ 2.0 when $s \in [2, 3]$. It guarantees that a disease with a confirmed unique ($s = 1$) feature reaches a minimum probability of 50% and therefore cannot be suppressed by noise from common symptoms.

The state of a diagnostic session at iteration t is described by the tuple (34)

$$R^{(t)} = \big(x^{(t)}, m^{(t)}, D^{(t)}_{\text{cand}}, \hat{y}^{(t)}\big). \tag{34}$$

Here $x^{(t)}$ is the current symptom vector, $m^{(t)}$ is the SOM membership, $D^{(t)}_{\text{cand}}$ is the candidate set, and $\hat{y}^{(t)}$ is the ranked list of diagnosis probabilities.

In Algorithm 1, SOM is used both for the initial clustering and at each subsequent iteration, where it dynamically updates the candidate set. When a new symptom is added, the patient's vector changes, which can change the BMU and, as a result, update the list of candidate diagnoses. The application layer also handles feedback after treatment. If the user reports that the suggested diagnosis did not explain the clinical course, the Feedback Processor revises the ranking. It lowers the confidence of that diagnosis, restores previously excluded hypotheses, widens α and k, and restarts the cycle. Cases in Table 9 terminated by criteria (b) or (c) rather than (a) explain the relatively low final probabilities (15.0% for influenza, 25.1% for meningitis). In these cases the top–1 ranking was stable across iterations even though the confidence threshold of criterion (a) was not reached.

5. Results of the study on disease clustering by symptom similarity for differential diagnosis

This section reports the results of evaluating the proposed technology. The subsections map onto the objectives as follows:

– Section 5.1 states the experimental protocol;

– Section 5.2 and 5.3 report the effect of normalization on clustering quality and the semantic coherence of the resulting clusters (objective 6);

– Section 5.4–5.6 report catalogue coverage, end–to–end diagnostic performance, and a case-level trace through the diagnostic cycle (objectives 7 and 4);

Algorithm 1. Dr.Case diagnostic cycle

Input: Initial symptoms S_0, disease database DB, trained SOM, neural network NN.
Output: Ranked list of diagnoses $\hat{y}$ with probabilities.

1. Initialize: Form the binary vector $x^{(0)}$ from the initial symptoms S_0.
2. L2-normalize: If $\|x^{(t)}\|_2 > 0$, compute $\hat{x}^{(t)} = x^{(t)}/\|x^{(t)}\|_2$. If $\|x^{(t)}\|_2 = 0$ (the user entered no symptoms or NLP recognized none), the system skips SOM projection and candidate selection (Steps 3 and 4), initializes a uniform prior $\hat{y}$ over the disease catalogue, and goes to Step 8 to select the first question from a predefined broad-screening symptom set or, in its absence, from the full symptom vocabulary using prior expected information gain.
3. Project onto the SOM: Find BMU = $\arg\min_u \|\hat{x}^{(t)} - w_u\|^2$ and compute the membership m_u for all units.
4. Select candidates: $D_{\text{cand}} \leftarrow$ union of diseases from units that cover 90% of the cumulative mass, with $k = 6$ as a lower bound and $\tau = 0.01$ as a cutoff.
5. Rank with NN: Compute $\hat{y} = \text{NN}(x^{(t)}, \text{BMU}_{\text{coords}})$ and filter the output to D_{cand}.
6. Apply Rule-based Boost and SymptomDrivenDiscovery to $\hat{y}$.
7. Check the stopping criteria: Return the current top-3 diagnoses if any of the following holds:
   (a) high confidence: $\hat{y}_{\text{top}} > 0.85$ and the gap between top-1 and top-2 probabilities exceeds 0.3;
   (b) no informative question remains: $\max_q \text{EIG}(q) < \varepsilon_{\text{EIG}}$ with $\varepsilon_{\text{EIG}} = 10^{-3}$;
   (c) stability: the identity of the top-1 candidate is stable for three consecutive iterations;
   (d) limit: the maximum number of iterations has been reached (see Step 12).
8. Select a question using FocusedQuestionSelector: $q^* = \arg\max_q \left[\text{EIG}(q) \cdot \text{bonus}(q)\right]$.
9. Obtain an answer: Ask the user about symptom q^* (Yes / No / Unknown).
10. Update the state: $x^{(t+1)} \leftarrow \text{update}(x^{(t)}, q^*, a)$ and apply BayesianAnswerProcessor to $\hat{y}$.
11. Repeat: Go to Step 2 with $t \leftarrow t + 1$.
12. Safety: With $T_{\max} = 20$, if $t \geq T_{\max}$, return the top-3 diagnoses for physician review.

– Section 5.7 reports the service performance of the software system (objective 5).

The theoretical result that underlies objective 1 is the bias of the unnormalized metric and its removal by normalization. This result is established in Section 4.2 (Theorems 1 and 2) and is referred to here only where it explains the measured effects.

5.1. Software system and its architecture

Based on the developed mathematical models and algorithms software system called Dr.Case was created. The Dr.Case system is organized as a four-layer software architecture, which separates the user-facing interfaces from the reasoning logic, the machine learning models, and the underlying data. Figure 1 shows the relationship between the layers.

The Presentation Layer exposes the system through two interfaces. The web UI is a Streamlit application with four pages (quick diagnosis, interactive session, database browsing, system information). The Representational State Transfer (REST) Application Programming Interface (API) is implemented with FastAPI and provides 15 endpoints for programmatic access, covering symptoms, diagnoses, sessions, and feedback. Pydantic models validate the inputs and outputs of every endpoint.

The Application Layer holds the domain logic of the diagnostic cycle. The Diagnosis Cycle Controller orchestrates the iteration by calling four components:

– the Session Manager, to maintain state;
– the Iteration Manager, to run one step of the inference loop;
– the Stopping Criteria checker, to decide when to terminate;

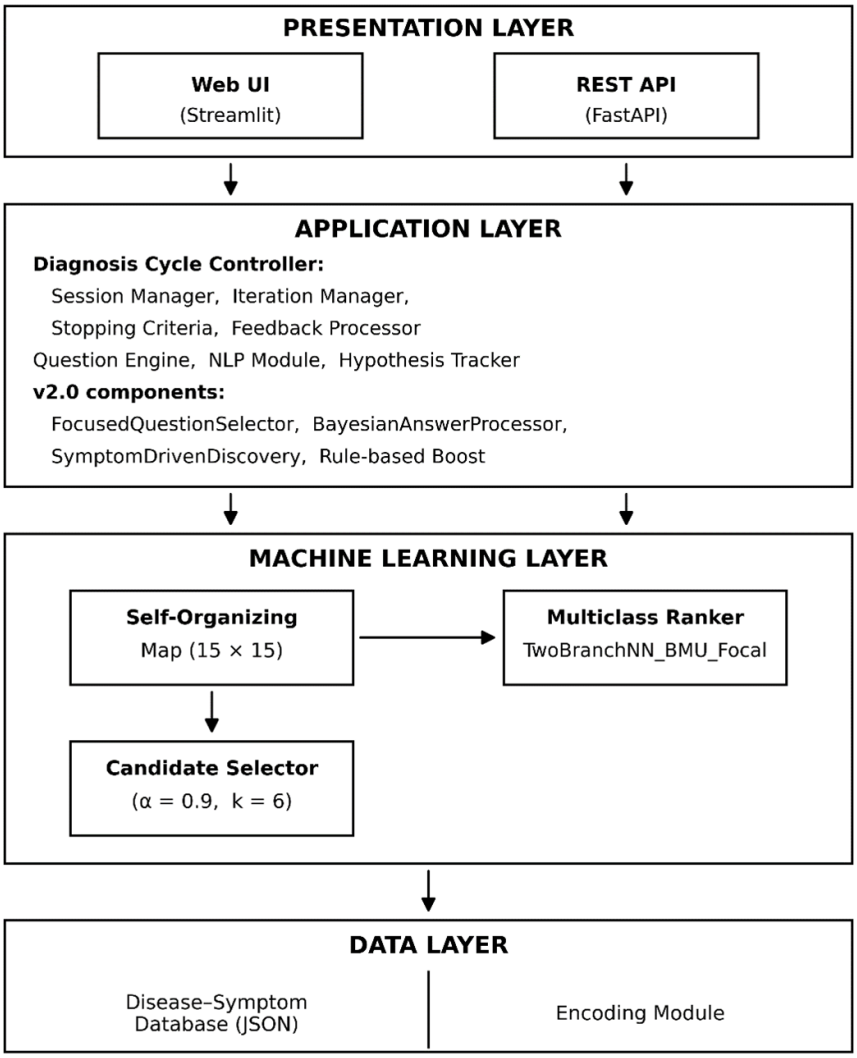

Fig. 1. Four-layer architecture of the Dr.Case software technology

– the Feedback Processor, when the user reports that a proposed diagnosis did not match the treatment outcome.

The Question Engine selects the next clarifying question. The NLP Module extracts symptoms from free-text complaints in Ukrainian and English. The Hypothesis Tracker records how the ranking evolves across iterations. The layer also contains components that implement specificity-aware reasoning.

The Machine Learning Layer contains the two learned models and the Candidate Selector that bridges them. The SOM clusters diseases by symptom similarity. The neural network ranks candidate diseases with context from the SOM. The Candidate Selector narrows the search space from 844 diseases to 20–50 candidates, with a self-projection recall of 99.5% on the full disease catalogue.

The Data Layer stores the disease–symptom database as JSON, together with the Encoding Module that converts symptoms and diseases into binary vectors and maintains the symptom vocabulary.

Modules are loosely coupled. The Machine Learning Layer depends only on the Encoding and Schemas layers, which allows SOM and neural network to be replaced by alternative implementations without changing the Application Layer. The Application Layer depends on the Machine Learning Layer through thin interfaces (project, rank, select) and does not reference specific model internals. Note: the source-tree directory `disease_ranker/` corresponds to the legacy path `multilabel_nn/`

used in earlier internal builds. The current implementation is a single-label multiclass ranker.

The system uses the following main libraries:

– NumPy and SciPy for numeric computation;
– PyTorch 2.0+ for deep learning;
– MiniSom for the SOM implementation;
– scikit-learn for auxiliary utilities;
– FastAPI with Uvicorn and Pydantic for the REST API;
– Streamlit for the web UI.

The total source base is more than 15,000 lines organized into more than 80 files. Minimum hardware requirements are 2 CPU cores and 4 GB RAM; recommended configuration is 4+ cores and 8+ GB RAM. A CUDA–capable GPU is used for neural network training but is not required at inference time.

The source code is partitioned into 16 functional modules arranged in a deterministic dependency order. Table 5 lists the modules and their responsibilities.

Table 5. Modules of the Dr.Case software technology

Module	**Responsibility**
config/	System configuration (default, optimized, runtime)
schemas/	Data structures (case records, vectors, iteration state)
encoding/	Vectorization of symptoms, diseases, patients
som/	Self-Organizing Map model, training, projection, index
candidate_selector/	Membership computation, selection policies, recall validation
pseudo_generation/	Synthetic patient cases for training the ranker
disease_ranker/	Two-branch disease-ranking neural network (single-label multiclass), training, inference, metrics
question_engine/	Expected Information Gain, question selection, answer processing
diagnosis_cycle/	Session, iteration manager, controller, stopping, feedback
validation/	Candidate recall, SOM quality, NN quality
optimization/	Tuners for SOM, selector, generator, NN, full pipeline
nlp/	Text preprocessing, symptom extraction, synonym dictionary, vitals
api/	FastAPI endpoints, Pydantic models, dependencies
web_ui/	Streamlit pages and components
data/	Database and serialized models
scripts/	Runner scripts for API, UI, and tests

API performance is also relevant for a software technology intended for real use. Preliminary local-load measurements were taken on a single workstation with sequential requests after model warm-up. Mean response times are approximately 12 ms for the quick-diagnose endpoint, 25 ms for the NLP symptom-extraction endpoint, and 8 ms for a session-step endpoint. On commodity hardware, the server sustains more than 100 requests per second. These numbers are reported as engineering indicators of feasibility, not as a formal benchmark, and a controlled multi-client load study with cold-start, warm-cache, and concurrent-user scenarios is left for future work.

5.2. Experimental setup and evaluation protocol

The evaluation is organized on four distinct levels.

1. SOM–level evaluation measures Quantization Error (QE), Topographic Error (TE), and Map Fill Ratio (Fill) for the trained self-organizing map on the full set of 844 disease vectors.

2. Selector self-projection evaluation projects each of the 844 disease vectors through the SOM and selector, and measures how often the disease itself appears in the resulting candidate set. This is a structural coverage check on the catalogue, not a test of generalization to unseen patients.

3. Model-level evaluation of the disease-ranking neural network uses a 75/15/10 train/validation/test split of synthetic patient cases generated from the disease database.

4. End–to–end case-level evaluation measures the overall behavior of the integrated diagnostic cycle on a small held-out demonstration set of 6 clinical cases. The set covers several categories of disease: infectious (influenza, COVID-19), neurological (meningitis), cardiological, gastroenterological, and respiratory. For each clinical case, an initial set of symptoms and the expected diagnosis were defined. Six cases support qualitative inspection of the diagnostic cycle on representative disease categories rather than a statistically powered estimate of accuracy.

The SOM was trained with the following parameters:

– map size: 15×15 units;
– training epochs: 1000;
– neighborhood function: Gaussian;
– initial learning rate: 0.5, with exponential decay to 0.01;
– initial neighborhood radius: 7.5, decaying to 0.5;
– initialization: PCA.

Parameters of the TwoBranchNN_BMU_Focal neural network:

– symptom branch [460 → 256 → 128],
– BMU branch [2 → 32 → 16],
– combined layer [144 → 128 → 844].

The loss function is Focal Loss [19] with $\gamma = 2.0$ and $\alpha = 0.25$. Label Smoothing [20] is applied with $\varepsilon = 0.1$. The optimizer is AdamW with learning rate 10^{-3} and Cosine Annealing Warm Restarts schedule. Batch size is 64, and training runs for up to 100 epochs with early stopping patience of 10.

5.3. Effect of normalization on clustering quality

Two SOM models were trained on the same data: one without normalization and one with L2 normalization. Both models had a grid of 15×15 units. The comparison is presented in Table 6.

Table 6. Comparison of SOM quality without and with L2 normalization

Metric	Without L2	With L2	Improvement
QE	2.79	0.82	2.79 → 0.82
TE	0.28	0.13	2.2 × better
Fill	37%	79%	+42 p.p.
Semantic cluster coherence	Low	High	Qualitative change

The experimental results are consistent with the theoretical expectations. Quantization error dropped from 2.79 without normalization to 0.82 with L2 normalization. Because L2 normalization changes the geometry of the input space (raw vectors have norm $\sqrt{k}$, while normalized vectors have norm 1). For this reason raw QE values before and after normalization are not directly comparable as absolute distances, and the numerical ratio $2.79/0.82 \approx 3.4$ cannot by itself be interpreted as a 3.4 fold improvement in clustering quality. The drop in QE should be read together with the reduction of topographic error (from 0.28 to 0.13) and the increase in map fill ratio. Taken together, these three indicators point to substantially improved SOM organization. After L2 normalization, the disease vectors are approximated by the SOM prototypes within the normalized input geometry. The lower TE indicates better preservation of the topological structure of the data.

The map fill ratio increased from 37% to 79%. Without normalization, many units remain empty because diseases with different numbers of symptoms project onto different regions of the map, leaving large unfilled areas. With L2 normalization, diseases are distributed more evenly, which gives more efficient use of the topological space of the map.

5.4. Semantic coherence of disease clusters

To confirm the semantic correctness of the clustering, the composition of the respiratory-disease clusters was analyzed. The results are given in Table 7.

Table 7. Clustering of respiratory diseases (with L2 normalization)

Disease	BMU	Nearest neighbors in cluster
Influenza	(13, 13)	Bronchitis, Common cold, Measles
COVID-19	(14, 13)	Influenza, Pneumonia, SARS
Pneumonia	(14, 12)	Bronchitis, COPD, Asthma
Common cold	(12, 14)	Pharyngitis, Laryngitis, Sinusitis
Tuberculosis	(12, 11)	Pneumonia, Lung cancer, Bronchiectasis

Analysis of the results demonstrates high semantic consistency of the obtained clusters. Influenza is located next to bronchitis and common cold, which matches clinical practice, where these diseases are often competing diagnoses. COVID-19 is placed near influenza and pneumonia, which is also clinically reasonable. Tuberculosis ended up in the same cluster as pneumonia and lung cancer, diseases that indeed share much of their symptom profile and require differential evaluation in practice.

For comparison, without L2 normalization a query with the symptoms “fever, cough, headache” returned a cluster with completely irrelevant diseases: oral mucosal lesion, poisoning due to gas, and smoking addiction. These diseases ended up in one cluster only because they had a similar number of symptoms, not a similar symptom set.

5.5. End-to-end diagnostic performance and catalogue coverage

Table 8 summarizes the overall test results for Dr.Case v2.0.

Table 8. End-to-end test results of the Dr.Case system

Metric	Value	Target	Evaluation set
Top-1 Accuracy	83.3% (5/6)	$\geq 70\%$	6 clinical cases
Top-3 Accuracy	83.3% (5/6)	$\geq 85\%$	6 clinical cases
Mean Reciprocal Rank (MRR)	0.875	≥ 0.80	6 clinical cases
Avg. questions to diagnosis	11.8	≤ 15	6 clinical cases
Candidate recall (SOM + Selector)	99.5% (840/844)	$\geq 99.5\%$	full 844 disease catalogue

The system reached 83.3% Top-1 accuracy on the case-level set, which exceeds the target value of 70%. The Top-3 accuracy is also 83.3%, which is below the planned target of 85% and is reported here as a target that was not reached on this six-case demonstration set. Top-1 and Top-3 accuracy coincide because the single failing case (out of six) did not place the expected diagnosis within the top three hypotheses. Widening the evaluation window from one to three therefore does not recover the missed case. The average number of clarifying questions was 11.8, acceptable for interactive diagnosis. The candidate self-projection recall of 99.5% (840 of 844 diseases) is a catalogue-coverage metric. For 840 of the 844 disease vectors, projecting the vector onto the SOM and running the selector yields a candidate set that contains the disease itself. This is a necessary but not sufficient condition for end-to-end recall on real patient inputs. Four diseases whose self-projection falls outside their own SOM neighborhood indicate borderline cases for the current index-building policy, and are a natural target for future refinement.

5.6. Case-level analysis of the diagnostic cycle

Consider in detail the diagnostic process for three representative cases (Table 9).

Table 9. Case-level trace through the diagnostic cycle

Disease	Initial rank	Questions	Final rank	Decisive component
Influenza	#2 (2.10%)	3	#1 (15.0%)	BayesianAnswerProcessor
COVID-19	#1 (10.1%)	1	#1 (86.8%)	Rule-based Boost
Meningitis	#1 (3.90%)	8	#1 (25.1%)	SymptomDrivenDiscovery

For COVID-19, the initial symptoms were fever, cough, fatigue, loss of smell. In the current database, "loss of smell" is a database-unique highly specific feature for COVID-19, in the 844-disease catalogue used in this study. Three probabilities appear at different stages of the pipeline for this case. The raw neural ranker assigned COVID-19 a probability of 2.5%. After candidate filtering by the Candidate Selector and renormalization over D_{cand}, COVID-19 appeared as rank #1 with 10.1%. This is the value shown in the "Initial rank" column of Table 9, which reports post-filtering probabilities. The Rule-based Boost then lifted COVID-19 to a probability floor of 50%. After a single clarifying question (shortness of breath = Yes), COVID-19 reached a confidence of 86.8% and became the final result.

5.7. Service performance of the software system

The case of meningitis was more difficult. The initial symptoms (fever, headache, neck pain) are not unique and appear in many diseases. The system asked 8 clarifying questions. The most informative one was the question about "neck stiffness", which in the current database is a highly specific feature for meningitis. SymptomDrivenDiscovery raised the floor probability for meningitis as soon as this feature was confirmed. After additional questions about nausea and vomiting, meningitis became the final diagnosis with a confidence of 25.1%.

6. Discussion of the results of the study on disease clustering by symptom similarity for differential diagnosis

The results of the experiments confirm the effectiveness of the proposed software technology for clustering states by feature similarity, based on SOM with L2 normalization. The main scientific contribution is the theoretical justification for the role of L2 normalization in the proposed Euclidean SOM pipeline, confirmed experimentally. A second contribution is the integration of the normalized SOM into a complete software system with explicit interfaces and well-defined modules.

L2 normalization was a necessary condition for the correct behavior of this particular Euclidean SOM configuration on binary data. Alternative configurations, such as cosine SOM, spherical SOM, normalized prototypes, or Jaccard or Ochiai distance, could remove the cardinality bias by other means and lie outside the scope of the present study. Without normalization in the Euclidean pipeline, pairwise distances mix the structural overlap of symptom sets with their cardinality, which leads to semantically incorrect clustering through SOM weight averaging. Four observations together support this conclusion:

- the drop of the quantization error from 2.79 to 0.82;
- the reduction of topographic error from 0.28 to 0.13;
- the increase of the map fill ratio from 37% to 79%;
- the semantic cluster inspection.

The same mechanism explains why the empirical gains are an induced rather than an exact effect. The equivalence between Euclidean distance and cosine similarity, proven in Section 4.2, holds exactly only for input-input comparisons, whereas the SOM prototypes are real-valued and not of unit norm. During training, normalization therefore reshapes the input geometry and produces the observed gains in topographic organization and map utilization rather than imposing cosine geometry directly.

The components (`FocusedQuestionSelector`, `BayesianAnswerProcessor`, `SymptomDrivenDiscovery`, `Rule-based Boost`) address specific failure modes of the base pipeline.

The `FocusedQuestionSelector` concentrates the search on questions that discriminate between the two leading hypotheses, weighted by the specificity of the symptom. The `BayesianAnswerProcessor` updates probabilities with likelihoods that depend on symptom specificity, which prevents common symptoms from swamping rare but diagnostic ones. The `SymptomDrivenDiscovery` component protects rare diseases from being hidden by the neural ranker when a highly specific symptom is reported. The `Rule-based Boost` adds an explicit deterministic path for well-known disease–feature associations from the medical literature. The case traces in Table 9 show that different components become decisive for different disease types. This pattern is consistent with the design intent of a layered system that combines several complementary mechanisms.

An important conceptual outcome of the work is the justification of clustering states by feature similarity instead of by traditional classifications. The fact that diseases with different etiologies (for example, influenza and COVID-19) fall into the same cluster is a desirable property of the system, rather than an error. Such diseases are competing diagnoses in differential diagnosis, and grouping them into one cluster corresponds to real clinical logic.

The results address the limitations identified in Section 2. Earlier self-organizing map studies were restricted to a single disease or a single class of diseases [3, 6, 7]. This restriction is lifted by building one clustering model over the full catalogue of 844 diseases and 460 symptoms. The reliance on quantitative indicators in place of binary symptoms [4, 6] is removed by operating directly on binary symptom profiles. The absence of a theoretical justification for the choice of metric and normalization on sparse binary data [5] is closed by Theorems 1 and 2. These theorems identify the cardinality bias of the unnormalized metric and prove that normalization reduces input distance to a function of cosine similarity alone. Modern symptom checkers and large language models offer limited interpretability [12, 15]. This work answers that limitation through the two-dimensional map, which exposes the clinical neighbourhood of each diagnosis (Table 7), and through the decisive component recorded for each case (Table 9).

General-purpose checkers tend to miss rare diseases [12, 16]. This study addresses that tendency through the candidate selector and the `SymptomDrivenDiscovery` component, which raise the floor probability of a rare disease once a highly specific symptom is confirmed. These results together provide the integrated software technology for clustering hundreds of states by feature similarity with subsequent iterative differential analysis whose absence motivated the study.

The system produces practically useful results even with a relatively small knowledge base of 844 diagnoses. This indicates that the approach is scalable and that it has the potential for application in real clinical settings after expansion of the database and validation on larger samples. The layered architecture also allows the medical knowledge base to be replaced by a database for a different domain, for example equipment fault signatures. Such a replacement requires no changes to the machine learning or application layers, provided that the state vectors remain binary.

Several limitations bound these conclusions. The end-to-end accuracy is measured on a demonstration set of six clinical cases. This set supports qualitative inspection of the diagnostic cycle rather than a statistically powered estimate of accuracy, and the Top-3 target of 85% was not reached on it. The candidate self-projection recall of 99.5% (840 of 844 diseases) is a catalogue-coverage metric. For 840 of the 844 disease vectors, projecting the vector onto the SOM and running the selector yields a candidate set that contains the disease itself. The technology has been validated in a single domain, medical differential diagnosis, and its applicability to other binary-state domains follows from the architecture but is not yet demonstrated. These points define the validation work required before clinical use and motivate the directions below.

Directions for future research. One direction is the development of a speech-input module for patient complaints. The integration of speech recognition and natural language processing will allow patients to describe symptoms in free form, which will improve the convenience of using the system. The

system will extract symptoms from the audio recording automatically and will form the input vector for diagnosis.

A second direction is the integration of multimodal data: laboratory test results, imaging data, and patient history. This will require extension of the neural network architecture and adaptation of the SOM for work with heterogeneous data.

A third direction is the application of Bayesian methods and Monte Carlo Dropout for uncertainty quantification of the diagnosis. Instead of a point probability estimate, the system could provide confidence intervals, an output that matters for medical applications.

Conclusion

The aim of the study has been achieved. The study developed a software technology for automated differential medical diagnosis that raises clustering quality, ranking accuracy, and the interpretability of results relative to existing symptom-based systems.

1. The choice of distance metric and the normalization of binary symptom vectors were justified theoretically. Two theorems establish in closed form that the unnormalized Euclidean metric on binary profiles is biased toward profile cardinality. They further establish that normalization reduces the pairwise input distance to a function of cosine similarity alone. Earlier medical self-organizing-map studies apply the metric without such analysis; the present work supplies the formal condition under which Euclidean comparison reflects structural symptom overlap rather than symptom count.

2. A clustering model of diseases by symptom similarity was built on a 15×15 Kohonen self-organizing map over normalized binary vectors for the full catalogue of 844 diseases and 460 symptoms. Unlike prior work limited to a single disease or a single class of diseases, the model represents the entire catalogue in one two-dimensional map.

3. A candidate-selection mechanism and a two-branch ranking neural network with Focal Loss and Label Smoothing were developed. The selector narrows the search space from 844 diseases to 20–50 candidates, and the network ranks them using both the symptom vector and the self-organizing-map context. This ranking stage is absent from learning-to-rank diagnosis systems that lack an unsupervised clustering layer.

4. An iterative diagnostic cycle was designed, combining Expected Information Gain question selection, specificity-aware Bayesian answer processing, and rule-based reinforcement for highly specific features. The case traces show that the decisive component differs by disease type, which raises the floor probability of rare diseases that frequency-based symptom checkers tend to miss.

5. The technology was implemented as the Dr.Case system, a four-layer software solution of 16 loosely coupled modules in Python 3.10+. The system provides a representational-state-transfer interface of 15 endpoints and a web user interface. Mean endpoint response times of 8–25 ms at more than 100 requests per second confirm feasibility for interactive use.

6. The improvement in clustering quality from normalization was confirmed on the 844-disease database. Quantization error fell from 2.79 to 0.82, topographic error from 0.28 to 0.13, and the map fill ratio rose from 37% to 79%. The two quantization-error values are measured in different geometries and are read as one consistent indicator among several rather than as a standalone ratio. Semantic coherence was validated: clinically competing respiratory diseases occupy adjacent map regions, while diseases with similar symptom counts but no shared manifestations are placed apart. This separation forms the basis of the system's interpretability.

7. The end-to-end behaviour was demonstrated on a held-out set of six clinical cases: On the six-case demonstration set, the system reached the following values: Top-1 accuracy 83.3% (5 of 6, target of 70% met), Top-3 accuracy 83.3% (target of 85% not reached), mean reciprocal rank 0.875, and an average of 11.8 clarifying questions. The candidate self-projection recall over the full catalogue is 99.5% (840 of 844 diseases). The six-case result supports qualitative inspection of the diagnostic cycle rather than a statistically powered estimate of accuracy.

The scientific novelty of the work is the formal condition, given by the two theorems, under which a Euclidean self-organizing map clusters sparse binary profiles by structural similarity rather than by feature count. A second element of the novelty is the use of this condition as the basis of an integrated technology for clustering hundreds of states with subsequent iterative differential analysis. None of the reviewed works provides such a technology. The practical significance is a reusable, interpretable decision-support technology for medical differential diagnosis. Through its layered architecture, the technology transfers to other domains in which a state is described by the presence or absence of binary features.

Acknowledgements

Declaration on the use of Artificial Intelligence. The authors used AI systems during the preparation of this manuscript for grammar and language editing. All AI-generated content was carefully reviewed and edited by the authors. The authors take full responsibility for the accuracy, integrity, and originality of the article.

Funding. This work was partially supported by the Erasmus+ Programme of the European Union Project Erasmus+, "The transferable training model – the best choice for training", 2024–2027, Algarve University, Portugal, University of Library Studies and Information Technologies, University of Bielsko-Biala, Poland, European Union.

References

[1] T. Kohonen, "Self-organized formation of topologically correct feature maps," *Biological Cybernetics*, vol. 43, no. 1, pp. 59–69, 1982. https://doi.org/10.1007/BF00337288.

[2] J. Vesanto and E. Alhoniemi, "Clustering of the self-organizing map," *IEEE Transactions on Neural Networks*, vol. 11, no. 3, pp. 586–600, 2000. https://doi.org/10.1109/72.846731.

[3] M. Nilashi *et al.*, "Coronary heart disease diagnosis through self-organizing map and fuzzy support vector machine with incremental updates," *International Journal of Fuzzy Systems*, vol. 22, no. 4, pp. 1376–1388, 2020. https://doi.org/10.1007/s40815-020-00828-7.

[4] L. Salvador-Carulla *et al.*, "Use of the self-organising map network (SOMNet) as a decision support system for regional mental health planning," *Health Research Policy and Systems*, vol. 16, no. 1, p. Art. no. 35, 2018. https://doi.org/10.1186/s12961-018-0308-y.

[5] Y. Zhang, T. Chai, and Z. Yang, "Self-organizing feature map for cluster analysis in multi-disease diagnosis," *Expert Systems with Applications*, vol. 37, no. 9, pp. 6359–6367, 2010. https://doi.org/10.1016/j.eswa.2010.02.074.

[6] P. K. Sharpe and P. Caleb, "Self-organising maps for the investigation of clinical data: A case study," *Neural Computing and Applications*, vol. 4, no. 4, pp. 219–229, 1996. https://doi.org/10.1007/BF01413710.

[7] M. A. Fekihal and J. H. Yousif, "Self-organizing map approach for identifying mental disorders," *International Journal of Computer Applications*, vol. 45, no. 7, pp. 25–30, 2012. https://doi.org/10.5120/6793-9120.

[8] M. Amos, J. Kaplan, K. Oliver, R. Mitchell, and G. Clarke, "Validating the knowledge represented by a self-organizing map with an expert-derived knowledge structure," *BMC Medical Education*, vol. 24, p. Art. no. 416, 2024. https://doi.org/10.1186/s12909-024-05352-y.

[9] Y. Miyachi, O. Ishii, and K. Torigoe, "Design, implementation, and evaluation of the computer-aided clinical decision support system based on learning-to-rank: collaboration between physicians and machine learning in the differential diagnosis process," *BMC Medical Informatics and Decision Making*, vol. 23, p. Art. no. 26, 2023. https://doi.org/10.1186/s12911-023-02123-5.

[10] L. A. Ferhi, M. B. Amar, F. Choubani, and R. Bouallegue, "Enhancing diagnostic accuracy in symptom-based health checkers: a comprehensive machine learning approach with clinical vignettes and benchmarking," *Frontiers in Artificial Intelligence*, vol. 7, p. Art. no. 1397388, 2024. https://doi.org/10.3389/frai.2024.1397388.

[11] Z. Zhang, Q. Zhang, L. Wang, and H. Liu, "Enhancing multi-label disease diagnosis through hypergraph clustering and multi-classification label entropy," *International Journal of Machine Learning and Cybernetics*, vol. 16, pp. 1753–1770, 2024. https://doi.org/10.1007/s13042-024-02447-2.

[12] Y. Harada, T. Sakamoto, S. Sugimoto, and T. Shimizu, "Longitudinal changes in diagnostic accuracy of a differential diagnosis list developed by an AI-based symptom checker: retrospective observational study," *JMIR Formative Research*, vol. 8, p. Art. no. e53985, 2024. https://doi.org/10.2196/53985.

[13] J. Knitza *et al.*, "Diagnostic accuracy of a mobile AI-based symptom checker and a web-based self-referral tool in rheumatology: Multicenter randomized controlled trial," *Journal of Medical Internet Research*, vol. 26, p. Art. no. e55542, 2024. https://doi.org/10.2196/55542.

[14] H. Fraser, D. Crossland, I. Bacher, M. Ranney, T. Madsen, and R. Hilliard, "Comparison of diagnostic and triage accuracy of Ada Health and WebMD symptom checkers, ChatGPT, and physicians for patients in an emergency department: Clinical data analysis study," *JMIR mHealth and uHealth*, vol. 11, p. Art. no. e49995, 2023. https://doi.org/10.2196/49995.

[15] J. M. Hoppe, M. K. Auer, A. Strüven, S. Massberg, and C. Stremmel, "ChatGPT with GPT-4 outperforms emergency department physicians in diagnostic accuracy: Retrospective analysis," *Journal of Medical Internet Research*, vol. 26, p. Art. no. e56110, 2024. https://doi.org/10.2196/56110.

[16] F. Brasil, L. A. Nardi, S. Fagherazzi, V. Boaretto, E. Ghidini, and E. M. De-Paris, "The impact of artificial intelligence in the odyssey of rare diseases," *Biomedicines*, vol. 11, no. 3, p. Art. no. 887, 2023. https://doi.org/10.3390/biomedicines11030887.

[17] J. Park, N. E. Kholy, Y. Kim, and S. Park, "XAI-based clinical decision support systems: A systematic review," *Applied Sciences*, vol. 14, no. 15, p. Art. no. 6638, 2024. https://doi.org/10.3390/app14156638.

[18] H. W. Loh, C. P. Ooi, S. Seoni, P. D. Barua, F. Molinari, and U. R. Acharya, "Application of explainable artificial intelligence for healthcare: A systematic review of the last decade (2011–2022)," *Computer Methods and Programs in Biomedicine*, vol. 226, p. Art. no. 107161, 2022. https://doi.org/10.1016/j.cmpb.2022.107161.

[19] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 2, pp. 318–327, 2020. https://doi.org/10.1109/TPAMI.2018.2858826.

[20] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the Inception architecture for computer vision," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 2818–2826. [Online]. Available: https://doi.org/10.1109/CVPR.2016.308

УДК 004.032.26:519.237.8

ПРОГРАМНА ТЕХНОЛОГІЯ ДЛЯ КЛАСТЕРИЗАЦІЇ СТАНІВ ЗА ПОДІБНІСТЮ ОЗНАК НА ОСНОВІ САМООРГАНІЗОВНИХ КАРТ КОХОНЕНА

Олексій Бичков
https://orcid.org/0000-0002-9378-9535

Максим Мельник
https://orcid.org/0009-0000-1180-9487

Катерина Меркулова
https://orcid.org/0000-0001-6347-5191

Володимир Петрівський
https://orcid.org/0000-0001-9298-8244

Київський національний університет імені Тараса Шевченка
Київ, Україна

Отримано: 19.04.2026р. / Прийнято: 11.05.2026р. / Опубліковано: 28.05.2026р.

У статті представлено програмну технологію кластеризації високовимірних станів за подібністю ознак на основі самоорганізовних карт Кохонена з L2 нормалізацією бінарних векторів ознак. Технологію реалізовано у вигляді багаторівневої програмної системи для автоматизованої диференціальної медичної діагностики. Наведено теоретичне обґрунтування етапу L2 нормалізації у формі двох теорем: перша виявляє систематичне зміщення ненормалізованої евклідової метрики щодо потужності бінарних профілів; друга показує, що L2 нормалізація усуває це зміщення та зводить попарну евклідову відстань між бінарними входами до функції виключно структурної (косинусної) подібності. На базі даних із 844 захворювань і 460 симптомів L2 нормалізація зменшує помилку квантування самоорганізовної карти з 2.79 до 0.82 (ці значення вимірюють відстані в різних геометріях і не є безпосередньо порівнюваними як абсолютні відстані), знижує топографічну помилку з 0.28 до 0.13 та збільшує коефіцієнт заповнення карти з 37% до 79%. Програмна система поєднує кластеризацію на основі самоорганізовних карт із модулем відбору кандидатів та двогілковою нейронною мережею ранжування захворювань, навченою із застосуванням *Focal Loss* і *Label Smoothing*, інтегрованими через ітеративний діагностичний цикл із вибором запитань на основі очікуваного приросту інформації, байєсівською обробкою відповідей з урахуванням специфічності та *rule-based* підсиленням для високоспецифічних ознак захворювань. Реалізацію організовано у вигляді 16 модулів Python із REST API та вебінтерфейсом користувача. Індекс самоорганізовної карти разом із модулем відбору кандидатів охоплює 99.5% каталогу з 844 захворювань у режимі самопроєкції (840 із 844 захворювань), а наскрізна система досягає точності Top-1 на рівні 83.3% на невеликому відкладеному демонстраційному наборі з шести клінічних випадків.

Ключові слова: системи підтримки клінічних рішень, диференційна діагностика, подібність ознак, самоорганізовні карти Кохонена, L2 нормалізація, архітектура програмного забезпечення.