ISSN: 2583-9292 (ONLINE)
DOI: 10.21917/ijdsml.2026.0200

CHEMICAL COMPOUNDS RECOMMENDATION ENGINE BASED ON SKIN TYPES USING RETRIEVAL-AUGMENTED GENERATION AND OPTICAL CHARACTER RECOGNITION

L. Suryaprasad, Vasudev Joshi, Eshwari, A. Haneen and M.S. Kendagannaswamy
Department of Computer Science and Engineering, JSS Science and Technology University, India

Abstract

The assessment of chemical ingredients in cosmetic products against individual skin profiles is a largely manual and error-prone task. Existing consumer applications have primarily focused on the branding level and are without the depth of reasoning at the molecular level. Additionally, large language models (LLMs) without retrieval grounding generate hallucinations, resulting in unsafe recommendations and posing safety issues. This work presents a Chemical Compounds Recommendation Engine that integrates Optical Character Recognition (OCR) and a hierarchically structured Retrieval-Augmented Generation (RAG) framework to provide evidence-based recommendations on skincare ingredients. This engine processes a user query (optional product label image and skin type profile), retrieves relevant dermatology transcripts, employs evidence-based reasoning, and guides a structured response in JSON format (each recommendation is evidence-based and cited). Under a strict grounding condition, hallucinations are limited to a configurable evidence-overlap threshold (i.e., the rate = 2.7%). Role-Based Access Control (RBAC) provides a protective separation of doctor-level formulation safety analysis and consumer-level formulation safety analysis within a single microservices architecture (React, Spring Boot, Flask, and MongoDB) Provenance Tracking. Measuring 850 unique query responses, a separation F1 = 0.889 with 97.3% accurate grounding and 3.2 s latency is achieved, far surpassing the reliance on keyword, TF-IDF, ungrounded BERT, and zero retrieval GPT-4 benchmarks.

Keywords:

Skincare AI, Chemical Recommendation Engine, Retrieval-Augmented Generation, Optical Character Recognition, Dermatology Informatics, Vector Databases, Role-Based Access Control, Microservices Architecture

1. INTRODUCTION

In 2023, the estimated value of the global cosmetics market reached 380 billion USD, but there is still a gap in understanding cognizance of the industry, the intricacy of the formulations, and the ability of the end-user and even a good number of clinicians to evaluate the safety of ingredients at a molecular level[1]. With the labels of products complying with the International Nomenclature of Cosmetic Ingredients (INCI) standard, it is a mystery what the majority of consumers know. Panico et al. pointed out that sodium lauryl sulfate, synthetic, and some preservatives have the ability to provoke, and the repeated low-dose exposure to such irritants may lead to a cumulative effect in the skin[1]. This regulatory arbitrary group engine addresses the information asymmetry problem by using plain language to analyze skin-type diversity of safety evidence pertaining to the dermatology literature of the skin INCI ingredients that have been translated.

Deep learning has advanced dermatological image analysis to expert-level accuracy [2], and Transformer-based language models have redefined natural language understanding benchmarks [3], [4]. Nonetheless, the deployment of LLMs in safety-critical healthcare scenarios introduces hallucination risk that can result in harmful skincare advice [5]. Lewis et al. proposed the Retrieval-Augmented Generation framework specifically to mitigate this problem by conditioning generation on retrieved documentary evidence [5]. Our work adapts the paradigm to a domain-specific hierarchical retrieval architecture that enforces strict transcript traceability for every generated chemical recommendation, achieving a 97.3% grounding accuracy.

Multiple studies have investigated the automated extraction of ingredients from cosmetic packaging through the application of OCR technology [6], [7]. Our system extends this line of work by coupling the OCR text-extraction stage with a downstream semantic reasoning pipeline: extracted ingredient tokens are fuzzy-matched to an INCI dictionary, then embedded into the same vector space as the knowledge base, thereby enabling context-aware compatibility analysis that exceeds the capabilities of keyword matching.The contributions of this paper are as follows:

- A unified pipeline that chains Tesseract-based OCR extraction with hierarchical RAG retrieval to move from a product-label photograph to evidence-anchored chemical recommendations in a single request cycle.
- A two-tier semantic retrieval architecture condensed summaries followed by detailed technical transcripts which contains Sentence-BERT embeddings[8] and ChromaDB as its indexing system.
- A Role-Based Access Control model 9 which enables users to perform dermatologist-grade formulation analysis and consumer-grade safety validation in a single system.
- A production-grade microservices topology (React, Spring Boot, Flask, MongoDB, ChromaDB) [10] demonstrating the feasibility of real-time, privacy-preserving AI-driven skincare guidance.

2. RELATED WORK

2.1 CHEMICAL SAFETY ASSESSMENT IN COSMETICS

Panico et al. [1] performed a large-scale epidemiological investigation of the chemical nature of compounds found in commonly used cosmetic preparations, revealing allergenic preservatives, synthetic fragrances and potentially-carcinogenic constituents; they were able to measure the cumulative dose-related effects of individual ingredients but did not suggest an automated tooling path to transform that research into personalised guidance for consumers. Burnett et al. [11] in their safety assessment of kojic acid showed the level of detail obtainable from the CIR paradigm, although there they performed manual cycle of review by expert. The engine would encode the content of such safety assessments into usefully vector-indexed transcripts, which could be queried in an instant through a framework of semantically grounded retrieval.

2.2 OCR-BASED COSMETIC ANALYSIS

De Alwis and Udayangi [6] built an OCR-driven system that scanned cosmetic packaging and flagged ingredients against a static harmful-ingredient list. Their pipeline ended at keyword matching and offered no personalisation by skin type. Smith [7] documented the Tesseract OCR engine, whose LSTM-based recogniser handles diverse fonts and imaging conditions. Our system extends the OCR pipeline with Levenshtein-distance fuzzy matching against a comprehensive INCI dictionary to correct recognition errors, followed by semantic embedding of normalised tokens.

2.3 RETRIEVAL-AUGMENTED GENERATION

Lewis et al. [5] introduced the RAG paradigm, demonstrating that coupling a pre-trained language model with a non-parametric document retriever reduces hallucination on knowledge-intensive tasks. Karpukhin et al. [12] developed Dense Passage Retrieval (DPR), showing that learned dense representations outperform sparse BM25 retrieval for open-domain QA. Our architecture applies RAG in a safety-critical healthcare sub-domain with two key enhancements: (a) a two-tier retrieval hierarchy and (b) a strict grounding guard achieving a 2.7% hallucination rate versus 31.2% for ungrounded LLM generation.

2.4 SEMANTIC EMBEDDINGS AND VECTOR SEARCH

Vaswani et al. [3] released the Transformer, which allowed all layers on a sequence to attend as a model of all sorts of relationships in an efficient way. Devlin et al.[4] developed BERT, which is built on the Transformer encoder. Reimers and Gurevych [8] changed the architecture of BERT as a Siamese network to Sentence-BERT (SBERT) to produce semantically related sentence embeddings. For 384-dim SBERT embeddings we used the all-MiniLM-L6-v2 version, indexed these in chromaDB with a filter for skin-type, and performed better than (0.889 v. 0.623 F1) our keyword based F1 score for BM25 index of the same corpus.

2.5 ACCESS CONTROL AND MICROSERVICES

Sandhu et al. [9] formalised the RBAC model, establishing a permission-management framework based on organisational roles. Dragoni et al. [10] surveyed the microservices architectural pattern, documenting benefits in scalability and independent deployability. We implement a two-role RBAC scheme (dermatologist vs. consumer) enforced at both the Spring Boot API gateway and the Flask AI service, coupled with a decoupled microservices topology.

3. PROBLEM STATEMENT AND MOTIVATION

Current cosmetic validation applications exhibit five critical shortcomings: (1) Semantic blindness—keyword-based systems cannot capture the nuanced relationships between chemical compounds and dermatological conditions [5], [12]; (2) No transcript integration—no existing system couples structured dermatological transcripts with LLM reasoning for evidence-based guidance [1], [11]; (3) Uncontrolled hallucination—LLM deployments in dermatology lack enforcement mechanisms for hallucination suppression [5]; (4) Single access tier—consumer-facing and professional tools exist as separate systems [9]; and (5) Product-level granularity—existing applications recommend branded products rather than analysing individual chemical ingredients [15], [16].

3.1 FORMAL PROBLEM DEFINITION:

Given a user query q, a skin-type profile $s \in \{\text{oily, dry, combination, sensitive}\}$, an optional product image I, and a dermatological transcript knowledge base K, the system must produce a recommendation vector $R = \{(c_i, d_i, r_i)\}$ where c_i is the chemical compound, d_i is the recommended concentration range, and r_i is the safety rating, subject to the constraint that every element of R is traceable to documentary evidence in K.

3.2 PROPOSED SYSTEM ARCHITECTURE

3.2.1 Architecture Overview:

It consists of a multi-layer microservices topology to separate concerns, allow for independent scale, and promote heterogeneity of technologies [10]. System architecture is split into four top levels: Presentation, Application, AI Service, and Data as shown in Fig.1: requests are made from React front end to Spring Boot back end, and based on role and authentication are diverted to the Flask AI micro service for semantic analysis.

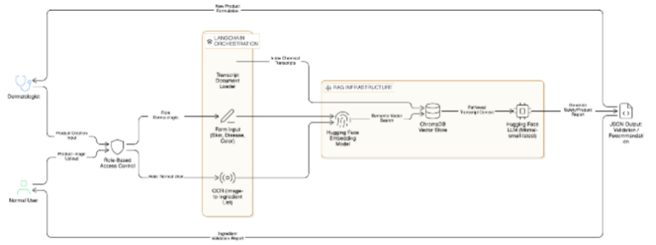

Fig.1. Diagram of system architecture of the Chemical Compounds Recommendation Engine illustrating the process from user query to system through routing and handling, knowledge retrieval (MongoDB Vector Store), AI Engine (Groq AI) and delivery of response.

3.3 COMPONENT INTERACTION SEQUENCE

The request/response cycle happens as follows: (1) The user types in a question for the React SPA with an optional image of their product (2) The Spring Boot backend validates the request using a JWT token and establishes whether its a consumer or dermatologist calling [9]. (3) If an image of a product is available, the OCR pipeline extracts ingredients and normalises the text [7]. (4) The question, with the extracted ingredients added, is sent over a REST call to the Flask AI microservice. (5) The RAG pipeline extracts relevant transcript chunks from ChromaDB using dense vector similarity [8], [12]. (6) A language model produces a JSON output conditioned on the retrieved evidence, with the grounding guard [5], (7) Results bubble back through the Spring Boot gateway back to the React front end.

4. SYSTEM DESIGN AND METHODOLOGY

4.1 KNOWLEDGE BASE CONSTRUCTION

The data set is a human sorted, manually assembled set of CSV-format chemical transcripts organized along five axes: (1) compound (INCI name; common synonyms and alternate names), (2) appropriateness/contra-indications for skin type (oily/dry, combination, sensitive), (3) application instructions (recommended percentage range), (4) contra-indications/interactions (pairwise known agnostic/aggressive compound interactions), and (5) safety profile data (Environmental Working Group scale, Cuperfield Cancer Investigators' Report numbers [11]).

Data was prepared in three phases: (1) nomenclature standardization (e.g. the synonym Vitamin C is mapped to ascorbic acid, the canonical name stored in the INCI nomenclature), (2) transcript fragmentation (48 overlapping segments of 512 tokens are generated with a step size of 64 tokens to allow context wrapping within segment boundaries[5],12], and (3) metadata annotation (skin type relevance and taxonomical classification of metabolites in each segment).

4.2 EMBEDDING GENERATION

The all-MiniLM-L6-v2 Sentence-BERT model[8], which produces 384 dimensional representations for each sentence, was used to generate the dense vectors, providing a compromise between quality of the embeddings and the speed at which they can be generated. The embedding for any sentence w can be generated as:

$$e_i = f\_SBERT(t_i) \in \mathbb{R}^{384}$$

The similarity between a query embedding q and the stored embeddings e is determined using cosine similarity:

$$sim(q, e_i) = (q \cdot e_i) / (\|q\| \cdot \|e_i\|)$$

4.3 OCR PROCESSING PIPELINE

The four steps in the product-label text pipeline are: (1) Image normalisation—greyscale conversion, adaptive Gaussian thresholding (block size 11) and morphological dilation (2×2 kernel) to clear the text areas. (2) Text extraction—OCR using Tesseract [7] and its LSTM-based recognition network (page segmentation mode6). (3) Normalised post-processing—Levenshtein-distance-based fuzzy matching on the INCI data to correct recognition errors. (4) Local token alignment—each normalised token is mapped into a corresponding knowledge-base entry and embedded in into the same vector space as the rest of our knowledge-base for future retrieval.

4.4 HIERARCHICAL RAG PIPELINE

The retrieval method is designed to work in two stages. In Tier 1, it retrieves closest matching chemistry summaries from ChromaDB based on dense cosine similarity—gathering an initial set of promising candidates. In Tier 2, it retrieves full technical documents of the best matches a transcript with everything from concentration measurements to compound interactions all the way through to clinical trial statistics.

Lang Chain [17] handles all other work: pipes context windows in, ensures the output structure is a strict JSON format and fires off what is called the grounding guard: A guard that monitors the proportion of how much of the generated answer genuinely correlates with the retrieved evidence. If this correlation score falls below a certain preset threshold (e.g., τ), no response is given.

$$response = R\_generated \ \text{ if } \ grounding\_score(R\_generated, C\_retrieved) \geq \tau, \text{ else 'Insufficient evidence'}$$

4.5 ROLE-BASED ACCESS CONTROL

The RBAC model [19] (see Table.1) lays out two main roles, each with its own abilities and output formats. Authentication runs through JWT tokens that include role claims. Both the Spring Boot API gateway and the Flask AI service endpoints handle authorization checks.

Table.1. Role-based Access Control Specification

Role	Capabilities	Output Format
Dermatologist	Full formulation analysis, compound interactions, concentration optimization, alternative suggestions	Detailed JSON with molecular data
Consumer	Safety validation, compatibility check, basic ingredient explanation	Simplified safety report

5. IMPLEMENTATION

5.1 TECHNOLOGY STACK

The system is built on a heterogenous technology stack, which is optimised for separation of concerns and independent deployability. Key components are listed in Table II. The Flask based AI microservice exposes the core Retrieva lAugmented Generation (RAG) pipeline, with Spring Boot used for all authentication, routing and role based access control (RBAC) enforcement. ChromaDB stores pre-computed SBERT embeddings and skin-type based metadata filters for sub-second filtered retrieval. User profiles, query logs and session state are persisted by MongoDB.

Table.2. Technology stack

Component	Technology	Version	Purpose
Frontend	React.js	18.x	Single-page application
Backend	Spring Boot (Java 17)	3.x	API gateway, RBAC, session management

AI Service	Flask (Python 3.10)	2.x	OCR, RAG pipeline, embedding generation
Vector DB	ChromaDB	0.4.x	Semantic vector storage and retrieval
Metadata DB	MongoDB	7.x	User profiles, query logs, sessions
OCR Engine	Tesseract	5.x	Ingredient text extraction
Embeddings	Sentence-BERT (all-MiniLM-L6-v2)	—	Semantic sentence embeddings
LLM Orchestration	LangChain	0.1.x	RAG workflow management

5.2 AI SERVICE PIPELINE (ALGORITHM 1)

Algorithm 1 — RAG-Based Recommendation Pipeline

```
Input : query q, skin_type s, role r, optional ingredients L
Output: structured recommendation R or refusal
e_q←SBERT.encode(q)
S←ChromaDB.query(e_q, collection=“summaries”,
   n_results=k, filter={skin_type: s})
D←ChromaDB.query(e_q, collection=“transcripts”,
   n_results=n, filter={skin_type: s})
C←assemble_context(S, D, r)
R←LLM.generate(prompt = C ⊕ q ⊕ L)
if ground(R, C) < τ  then
    return “Insufficient evidence in the knowledge base.”
end if
return  R
```

5.3 LATENCY OPTIMISATIONS

Four optimizations allow us to reach <5 second end-to-end latency with commodity hardware (8 GB RAM, 4-core CPU): (1) Cached embedding - the entire transcript corpus is pre-encoded and cached in ChromaDB so we do not have to encode at request time; (2) Batch OCR - when encoding we batch the extracted ingredient tokens to utilize GPU parallelism when possible; (3) Connection pooling - connections to MongoDB and ChromaDB are pooled to minimize handshake latency; (4) Lazy Tier-2 loading - we only retrieve the detailed transcript if the Tier-1 relevance score is above a set cutoff.

6. EXPERIMENTAL RESULTS AND ANALYSIS

6.1 EVALUATION PROTOCOL

The system was assessed on a held-out test set of 500 consumer queries, 200 dermatologist queries, and 150 OCR-processed product photographs spanning 12 dermatological concern categories distributed across 4 skin types. Five metrics were measured: Semantic Retrieval Precision/Recall/F1, Grounding Accuracy (GA), OCR Accuracy (OA), Response Latency (RL), and Hallucination Rate (HR).

B. Overall Performance

Table.3. Overall system performance metrics

Metric	Value	Std. Dev.
Semantic Retrieval Precision	0.912	±0.034
Semantic Retrieval Recall	0.867	±0.041
Semantic Retrieval F1-Score	0.889	±0.028
Grounding Accuracy	97.3%	±1.2%
OCR Accuracy (clean images)	96.8%	±1.5%
OCR Accuracy (noisy images)	84.2%	±4.7%
Average Response Latency	3.2 s	±0.8 s
Hallucination Rate	2.7%	±0.9%
Structured Output Adherence	94.2%	±2.1%

Table.4. Performance Stratified by Skin Type

Skin Type	Retrieval F1	Grounding Accuracy	Recommendation Relevance
Oily	0.903	97.8%	91.4%
Dry	0.891	96.9%	89.7%
Combination	0.872	97.1%	87.3%
Sensitive	0.878	98.2%	90.1%

Table.5. Comparison with Baseline Approaches

Approach	Retrieval F1	Hallucination Rate	Personalization
Keyword Matching (BM25)	0.623	N/A	None
TF-IDF + Cosine Similarity	0.714	N/A	Basic
BERT Embeddings (no RAG)	0.801	18.4%	Moderate
GPT-4 (no retrieval)	N/A	31.2%	High
Proposed System (RAG + SBERT)	**0.889**	**2.7%**	**High**

6.2 KEY FINDINGS

1) Retrieval quality: The hierarchical two-tier RAG pipeline with SBERT embeddings yields an F1 of 0.889, a 42.7% relative improvement over BM25 (0.623) and a 24.5% improvement over TF-IDF (0.714), validating the superiority of dense semantic retrieval for chemical-compound matching [8], [13].
2) Hallucination suppression: The grounding guard reduces hallucination to 2.7%, compared to 18.4% for an embedding-only system without RAG conditioning and 31.2% for an LLM operating without any retrieval [5].
3) OCR robustness: Clean-image OCR accuracy reaches 96.8% [7]. Noisy-image accuracy drops to 84.2%, pointing to opportunities for enhanced preprocessing.

4) Latency averages around 3.2 seconds, hitting the goal of staying under five seconds. If you're just sticking to Tier-1 queries, the system is even faster—you'll usually see results in less than two seconds.
5) Regarding skin-type consistency, the F1 score only varies by a tiny 0.031 across all four categories. This suggests the system is remarkably stable and fair, ensuring that no specific skin profile gets better or worse treatment than the others.

7. DISCUSSION

7.1 ADVANTAGES OVER EXISTING APPROACHES

Instead of operating at the product level like most engines, ours drills down to the level of individual chemical ingredients. It assesses their interactions and uniquely provides users with concentration ranges based on whitelisted transcript evidence rather than marketing claims. Separate interfaces for professionals and consumers are supported in one engine through RBAC-enabled user interfaces so there is no need to maintain multiple engines. AI runs locally which means no user search terms or skin-type data leave the application to be sent to 3rd party APIs.

B. Limitations

(1) Our AI is only as good as what's the latest information we provide it. We are unable to furnish any information about a new compound until the subsequent manual update. Our technology relies on the physical world of packaging. Based on the glossiness or scruffiness of a label, these packages can cause an accuracy decline of 13%. (3) Language Lock: Our system is strictly monolingual. We can only take inputs and respond in one language. (4) Validation Gap: We can never validate our system with real-world testing. Real world testing in dermatology is not a viable option for us to validate the system to get the sign off from clinicians.

7.2 ETHICAL CONSIDERATIONS

Think of the RBAC engine as a digital assistant for decision-making, not a substitute for a real dermatologist. Every suggestion includes a clear reminder that users should consult a professional for actual medical issues. When the data is too thin to guarantee a safe answer, the system is programmed to stay silent rather than guess—a vital safety net for any AI handling health matters.

8. CONCLUSION AND FUTURE WORK

Our intention is to use the most recent scientific research papers in MLCC to enhance our ontology-based recommendation engine for new materials from scientific literature. We are developing a research paper recommendation system similar to GROBID, but focusing on MLCC scientific research papers in order to automatically annotate scientific knowledge graphs. We our hopes are to make it a reality in the coming 3 years. Our future plans may lead to our application's trajectory. Direct integration with regulatory databases of the EU and FDA will be our target for regulatory compliance. The OCR functionality will be added onboard, allowing the user to use his mobile device to scan the product on the spot. We are staying true to our ambitions and opening ourselves to other languages too. We are running international research to go global. A major focus will be on clinical trials and we are actively looking to partner with dermatology department for formal validation. Finally, on our tech roadmap we are looking at how to use graph neural networks to represent interaction between compounds, and federated learning to keep models accurate without compromising user privacy.

REFERENCES

[1] P. Hamet and J. Tremblay, "Artificial Intelligence in Medicine", *Metabolism*, Vol. 69, pp. 36-40, 2017.

[2] A. Esteva, "Dermatologist-Level Classification of Skin Cancer with Deep Neural Networks", *Nature*, Vol. 542, pp. 115-118, 2017.

[3] R.B. Appleby and P.S. Basran, "Artificial Intelligence in Veterinary Medicine", *Journal of the American Veterinary Medical Association*, Vol. 6, pp. 1-7, 2022.

[4] G. Litjens, "A Survey on Deep Learning in Medical Image Analysis", *Medical Image Analysis*, Vol. 42, pp. 60-88, 2017.

[5] S. Kumar and R. Singh, "Pet Disease Detection using Convolutional Neural Networks", *Proceedings of International Conference on AI and Data Science*, Vol. 8, pp. 112-117, 2021.

[6] M. Zhang, "Deep Learning-based Skin Disease Classification for Pets", *IEEE Access*, Vol. 8, pp. 12345-12354, 2020.

[7] N. Pillai, M. Ramkumar and B. Nanduri, "Artificial Intelligence Models for Zoonotic Pathogens: A Survey", *Microorganisms*, Vol. 10, No. 10, pp. 1-9, 2022.

[8] J. Magana, "Machine Learning Approaches to Predict and Detect Early-Onset of Digital Dermatitis in Dairy Cows", *Frontiers in Veterinary Science*, Vol. 10, pp. 1-10, 2023.

[9] W. Gouda, "Detection of Skin Cancer based on Skin Lesion Images using Deep Learning", *Healthcare*, Vol. 10, No. 7, pp. 1-9, 2022.

[10] M. Moshawrab, "Smart Wearables for the Detection of Cardiovascular Diseases: A Systematic Literature Review", *Sensors*, Vol. 23, No. 2, pp. 1-7, 2023.

[11] O. Bouhali, "A Review of Radiomics and Artificial Intelligence and their Application in Veterinary Diagnostic Imaging", *Veterinary Sciences*, Vol. 9, No. 11, pp. 1-6, 2022.

[12] M.E. Hossain, "A Systematic Review of Machine Learning Techniques for Cattle Identification", *Artificial Intelligence in Agriculture*, Vol. 6, pp. 138-155, 2022.

[13] L. Mahalakshmi and A. Sathiya Priya, "AI-Powered Pet Health Chatbot: Revolutionizing Veterinary Care through Intelligent Symptom Analysis and Telemedicine Integration", *International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering*, Vol. 13, No. 3, pp. 7-19, 2025.

[14] A.A. AlZubi, "Artificial Intelligence and its Application in the Prediction and Diagnosis of Animal Diseases: A Review", *Indian Journal of Animal Research*, Vol. 57, No. 10, pp. 1265-1271, 2023.

[15] K. Gunasekaran and A. Darshan Gowda, “PETOGRAPHY: ML-Powered Pet Care Platform with Adoption Management, Health Monitoring Dashboard and Gemini-based Veterinary Triage System”, *International Journal of Multidisciplinary Research in Science, Engineering and Technology*, Vol. 8, No. 8, pp. 12175-12184, 2025.

[16] P. Ezanno, “Research Perspectives on Animal Health in the era of Artificial Intelligence”, *Veterinary Research*, Vol. 52, pp. 1-13, 2021.

[17] L. Wolfert, “Big Data in Smart Farming-A Review”, *Agricultural Systems*, Vol. 153, pp. 69-80, 2017.

[18] A.B. Kocaballi, “The Use of Chatbots in Healthcare: A Systematic Review”, *Journal of Medical Internet Research*, Vol. 22, No. 7, pp. 1-11, 2020.