IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 1, January 2024

DOI: 10.17148/IJARCCE.2024.13104

DEVELOPMENT OF SMART PATIENT-CENTERED HEALTH CONSULTANCY MODEL: USING INTERNET OF THINGS BASED MULTI-AGENT

Daniel Khaoya Muyobo[1], Paul Oduor Oyile[2]

Department of Information Technology, Kibabii University, Kenya[1,2]

Abstract: The Internet of Things (IoT) has revolutionized healthcare by enabling the collection of vital patient symptoms and data from various devices. This data-driven approach forms the basis for a multi-agent model called Smart Patient-Centered Health Consultancy Model (SPCC-Model). This paper explores the constructs and sub-constructs of the SPCC-Model, encompassing IoT technology, technological infrastructure, government policies, multi-agent systems, analysis of patient data, classification analysis, and cause-effect analysis. This model integrates IoT technology, data analysis, healthcare providers, and patient preferences to deliver personalized and timely healthcare services. The model incorporates data collection, preprocessing, feature extraction, predictive analysis, continuous monitoring, and user interaction to provide personalized healthcare services. By leveraging IoT data, the SPCC-Model aims to enhance patient-centered care, improve healthcare outcomes, and ensure compliance with healthcare regulations.

Keywords: Data Analysis, IoT Devices, Healthcare, Multi-Agent Model, Patient-Centered Care, Predictive Analysis, Healthcare Technology, model validation.

I. INTRODUCTION

The development process of a multi-agent model for smart patient-centered health consultancy in health brings together all technologies that gives a single end-to-end integrated solution for healthcare. The modeling process which is a high-level overview of a multi-agent model including patient Agent, healthcare provider agent, care coordinator agent, data analytics agent and security agent. In real applications, models are presented in many shapes, sizes, and styles [1]. A model is not the real world but merely a human construct that simplifies the understanding of the real world systems. System modeling is the process of developing abstract models of a system, with each model presenting a different view or perspective of that system. Models of the new system are used during requirements engineering to help explain the proposed requirements to other system stakeholders. It represents the system from different perspectives. These perspectives include; an external perspective, where the context or environment of the system is modeled, an interaction perspective where the interactions between a system and its environment or between the components of a system is modeled, a structural perspective, where the organization of a system or the structure of the data that is processed by the system is modeled and a behavioral perspective, where the dynamic behavior of the system and how it responds to events is modeled [2].There are three ways in which graphical models are commonly used that is as a means of facilitating discussion about an existing or proposed system, as a way of documenting an existing system and as a detailed system description that can be used to generate a system implementation.

The study utilized the principles of modeling as applied in software development, which lay the foundation for a systematic approach to problem-solving and solution development. The first principle underscores the critical impact of model selection on problem interpretation and solution formulation, emphasizing the need for models that expose key development challenges. Precision levels in modeling, the second principle, recognize the varying needs of different stakeholders, allowing models to be adapted to levels of detail suitable for analysts, end-users, and developers. The third principle stresses the necessity of bridging the gap between analysis and design models, underscoring the importance of models closely aligned with the reality of the system. Embracing a multi-model strategy, the fourth principle acknowledges that a singular model cannot fully encapsulate complex systems, advocating for a set of interconnected models to provide comprehensive insights. These principles collectively provide valuable guidelines for effective software development, enhancing the understanding of problems and the crafting of well-rounded solutions [2].

Common modeling diagrams are used, example UML class diagrams are used when developing an object-oriented system model to show the classes in a system and the associations between these classes. An object class can be thought of as a

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 1, January 2024

DOI: 10.17148/IJARCCE.2024.13104

general definition of one kind of system object. An association is a link between classes that indicates that there is some relationship between these classes. When developing models during the early stages of the software engineering process, objects represent the real world, such as a patient, a prescription, doctor, etc. Figure 5.1 provide an ontological UML diagram for consultation process.

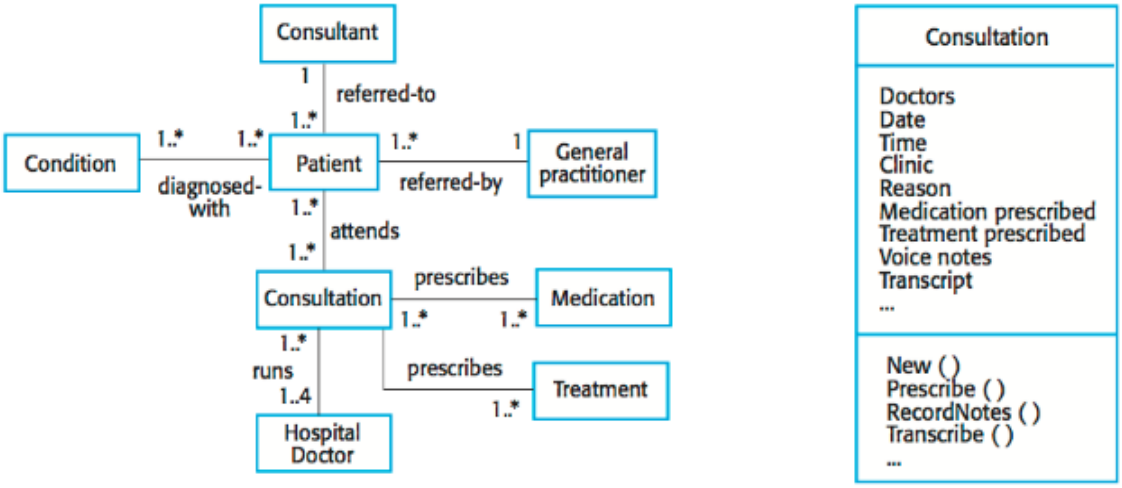

Fig. 1: UML Diagram

Fig. 1 is a Structural models displaying the health care system in terms of the components and their relationships. This serves as prototype to health care consultancy services.

The methodology for implementing the SPCC-Model involved several key steps including (i) Identifying Stakeholders and Requirements, including healthcare professionals, caregivers, and patients. Understand their requirements and preferences for healthcare services (ii) IoT Selecting and integrating IoT devices for data collection, such as wearables and medical sensors. Develop data preprocessing and cleaning techniques to ensure data quality (iii) Utilizing data analysis techniques, including statistical and machine learning models, for predictive analysis. Train models using historical patient data to predict health outcomes and assess disease risk (iv) Defining agent roles and responsibilities, communication protocols, and collaboration mechanisms. Emphasize patient-centered care and real-time monitoring (v) Ensuring compliance with government policies and legal systems related to data privacy, security, telemedicine, and health data interoperability and (vi) Validating the SPCC-Model through simulations, real-world data tests, and user feedback

II. ALGORITHM FOR PATIENT REQUEST AND SERVICE DELIVERY

The study devised an algorithm below in order to simulate the model development process based on the requirements specified as below:

Step 1: Initializing system components

i. Setting up a database or storage bank to store patient historical data, including vital signs and service requests.
ii. Establishing a communication platform for the patient to interact with healthcare providers and support services.
iii. Connecting sensor devices to the patient to collect vital signs data.

Step 2: Starting Simulation Loop (Repeating until simulation end conditions are met):

a. Simulate patient behavior:
i. Generate a random event representing the patient's request for a specific service.
ii. Collect vital signs data from the connected sensor devices.
iii. Store the collected data in the patient's historical data record.

b. Process the patient's request:
i. Retrieve the patient's historical data to determine the frequency and priority of the requested service.
ii. Evaluate the availability of the requested service and the resources needed to deliver it.

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 1, January 2024
DOI: 10.17148/IJARCCE.2024.13104

c. Deliver the service: (If the service is available and resources are sufficient):

i. Notify the patient about the service's acceptance and the estimated delivery time.
ii. Allocate the necessary resources to deliver the service.
iii. Update the patient's historical data to reflect the service delivery.
If the service is unavailable or resources are insufficient:
iv. Notify the patient about the unavailability of the requested service.
v. Provide alternative options or suggest consulting a healthcare consultant for further assistance.

d. Simulate healthcare consultant involvement:

i. If the patient requires further assistance or guidance, involve a healthcare consultant (expert system).
ii. The healthcare consultant reviews the patient's historical data and provides recommendations or modifies the requested service based on their expertise.
iii. Update the patient's historical data to reflect the consultant's input.

e. Simulate communication with support services:

i. If necessary, interact with other healthcare support services (e.g., lab tests, specialist consultations) to fulfill the requested service.
ii. Communicate with the support services through the established communication platform.
iii. Update the patient's historical data to reflect the support services' involvement.

Step 3: End simulation.

A. **Basic Patient Symptom Captured During Consultation Process**

The study finding through an intensive interviews with a medical experts identified the following patient data that were very core when requesting for healthcare service or consultation process. This finding as summarized in Table 1 as patient symptoms and data captured.

Table 1: Patient Symptoms and Data Captured in Consultation Process

Patient Data	Explain as per Medical Expert(s)	Basic Samples
Chief Complaint	The main reason for the patient seeking medical attention	• Headache • Cough • abdominal pain
Presenting Symptoms	A detailed description of the symptoms the patient is experiencing, including their duration, severity, and any associated factors. (Such as nausea or sensitivity to light) would are noted.	if a patient presents with a headache the: • location • intensity • duration • triggers • accompanying symptoms
Medical History	Gathering information about the patient's past medical conditions.	• Surgeries • allergies • family medical history
Review of Systems	Assessing various body systems to identify any additional symptoms the patient may be experiencing.	Asking the patient about the: • respiratory system • cardiovascular system • gastrointestinal system • musculoskeletal system
Physical Examination Findings	Conducting a physical examination to assess vital signs (blood pressure, heart rate, temperature),	• Observe any visible abnormalities and check specific body regions related to the presenting symptoms.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 1, January 2024

DOI: 10.17148/IJARCCE.2024.13104

		• Physician may palpate the abdomen or listen to the lungs using a stethoscope.
Past Medications and Treatments	Inquiring about any medications or treatments the patient has previously taken for the current condition or other health issues	• Drugs that were given
Social and Lifestyle Factors	Understanding the patient's lifestyle, occupation, living environment, habits	• smoking • alcohol consumption
Allergies and Drug Reactions	Identifying any known allergies or adverse reactions to medications or substances	• if allergic to some drugs
Associated Symptoms	Exploring any additional symptoms that may be related to the primary complaint or suggest an underlying condition	• weight loss • fatigue • changes in appetite • sleep disturbances

Table 1 provides a summary of relevant information captured by a consultant from the patient during consultation process. This finding were considered to be core in initiating decision making process, communication and storage using the data and requirements collected for technological mapping, frameworks, and machine learning algorithms.

B Technological Tools used in Capturing Patient Symptoms/Data

The study identified the following technological tools being used in capturing patient symptoms during a consultation as in Table 2.

Table 2: Data Capturing and Storage Tools in Healthcare

Tool	Explanation	Data captured	General function
Electronic Health Records (EHR)	EHR systems allow healthcare providers to record and access patient information electronically	Providers can input and update symptoms: • medical history • relevant details during consultations	Helps in maintaining accurate and organized patient records
Symptom Checkers	Online or mobile-based symptom checkers can help patients describe their symptoms and provide preliminary assessments	These tools use algorithms and databases to offer possible diagnoses or recommendations	helpful for patients to gather information, they should not replace professional medical advice
Wearable Devices and Sensors	Various wearable devices and sensors, such as fitness trackers or smartwatches, can monitor certain health parameters	capture data on heart rate, sleep patterns, activity levels, and even some physiological indicators	information can be shared with healthcare providers to aid in the assessment and management of symptoms
Telemedicine Platforms	Telemedicine platforms enable remote consultations between patients and healthcare providers	Through video or audio calls, patients can describe their symptoms to the provider, who can then document the information in the patient's medical record	Some platforms also offer features for patients to input their symptoms or medical history before the consultation.

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 1, January 2024
DOI: 10.17148/IJARCCE.2024.13104

Mobile Apps and Surveys	Mobile applications and surveys designed for symptom tracking can be used to capture and monitor specific symptoms over time	Patients can input information about their symptoms, their severity, triggers, and other relevant details	This data is valuable for both patients and healthcare providers to understand symptom patterns and make informed decisions

The study went further and identified and provide the description of the some specific technological tools that had capability to capture patient symptoms as in Table 3.

Table 3: Specific Patient Data Capturing Tools

Tool	Description and Function
Epic Systems	Epic is a widely used electronic health records (EHR) system that allows healthcare providers to document patient symptoms, medical history, and other clinical information during consultations.
Isabel Symptom Checker	Isabel is an online symptom checker that helps patients and healthcare providers identify potential diagnoses based on entered symptoms and clinical information
Fitbit	Fitbit is a popular wearable device that tracks various health parameters, including heart rate, sleep patterns, and activity levels. The collected data can be shared with healthcare providers to provide insights into a patient's overall health and potentially capture relevant symptoms.
Doximity Dialer	Doximity is a telemedicine platform that offers a secure video calling feature called Doximity Dialer. It allows healthcare providers to have remote consultations with patients and document symptoms and other relevant information during the call
HealthKit (Apple) and Google Fit (Google)	HealthKit and Google Fit are platforms for iOS and Android devices, respectively, that integrate data from various health and fitness apps. These platforms can capture information such as symptoms, activity levels, sleep patterns, and vital signs from compatible apps and devices.
Symple	Symple is a mobile app that allows patients to track and record their symptoms, medications, and other health-related data. It provides customizable questionnaires and visualizations to help patients and healthcare providers monitor symptom patterns
MyChart	MyChart is a patient portal that connects patients with their healthcare providers. It often includes features for patients to input symptoms, update medical history, and communicate securely with their providers

As discussed in the Table 3, these devices are integrated with IoT paradigms and gateway levels. The IoT paradigm comprise of numerous IoT devices with associate healthcare sensors and actuators. Typical healthcare sensors in the IoT paradigm include Blood pressure sensors, Pulse Oximeters as well as Electrocardiogram, Body temperature and Airflow sensors. Such sensors are geographically distributed and perceive the external environment to make decision and providing utilities for the corresponding system. Inherently, the devices are energy constrained and perform minimal computational tasks although they may generate a large amount of data within a short period of time. Therefore, to process the IoT data and host relevant applications, smart systems employ sophisticated deep learning models which range from Convolutional Neural Networks (CNNs) to sequence models like Long-Short-Term-Memory (LSTM) networks 16. The model also extends infrastructure, platform and software services to Edge and Cloud computing. In this case, the sensed data signals and functional requirements such as expected accuracy rate, data sensing frequency and service delivery deadlines are forwarded to computing paradigms from the IoT domain.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 1, January 2024

DOI: 10.17148/IJARCCE.2024.13104

The gateway level devices include smartphones, mobile computers like laptops and tablets acting as an interface between the IoT layer and computational resources. These devices collect data and package them into tasks to forward to the IoT-Fog broker. These devices can also be configured to manage data (to a small extent) to meet the user specifications or application Quality of Service (QoS) characteristics. Designing the software modules for such devices is challenging because one gateway device might be connected to multiple IoT devices or sensors and need to manage data and service requirements for each sensor or user. Further, these devices are prone to diverse security threats and malicious attacks that might steal data or sabotage one or multiple such gateway devices. Different techniques can be used to mitigate these difficulties like virtualization, encryption, blockchain based data management with close interaction with Broker and Worker nodes to allow more robust and seamless communication among the IoT devices, gateway nodes and Broker nodes 17. Figure 3 provide a prototype that visualizes the gateway level.

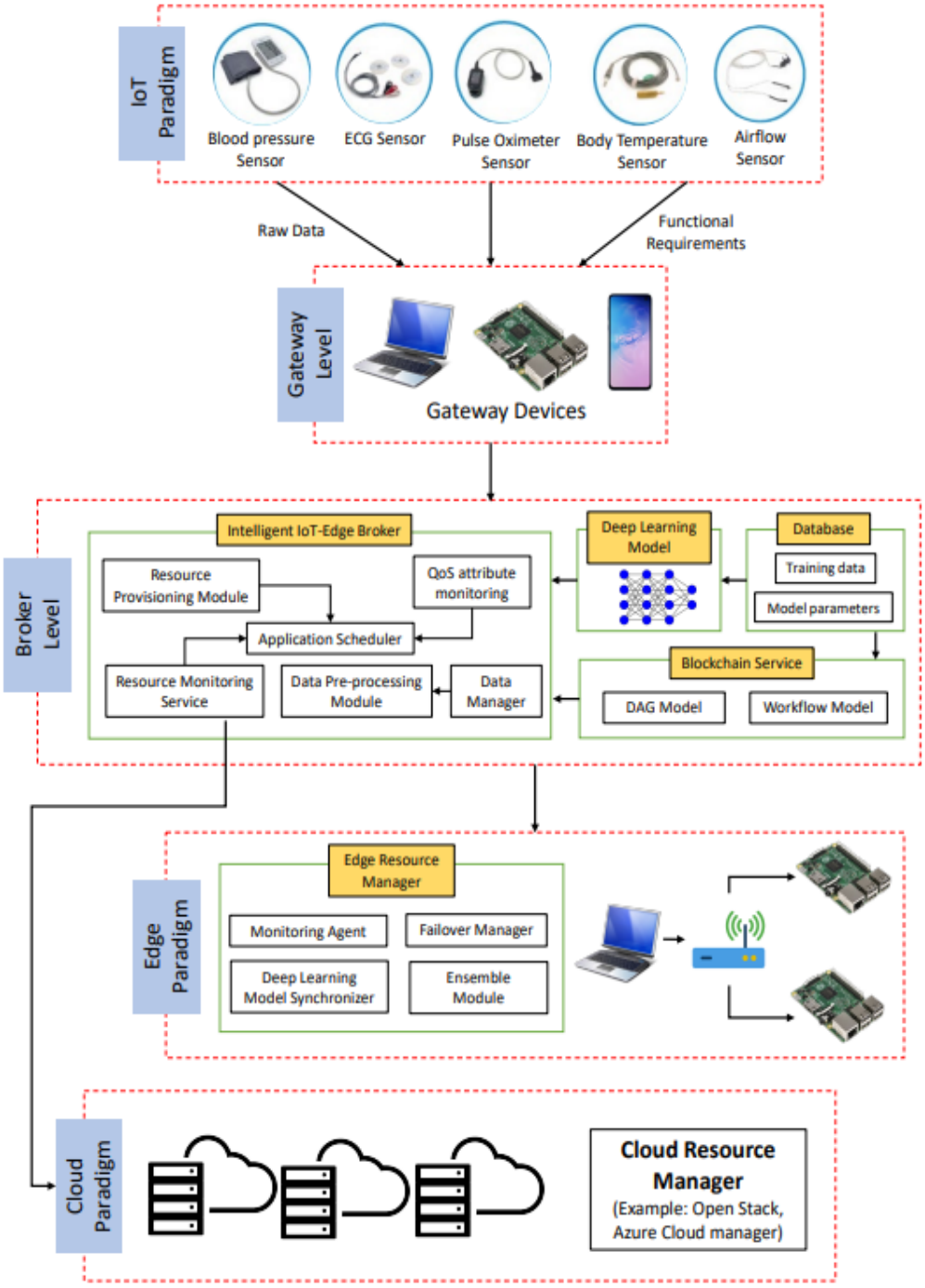

Figure 3: Visualizes the Gateway Level

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 1, January 2024

DOI: 10.17148/IJARCCE.2024.13104

B. **Architecture of the Algorithm**

Architecture of the algorithm that simulate the integration of various technological tools for smart patient-centered healthcare services was represented in a high-level diagram as in Figure 4.

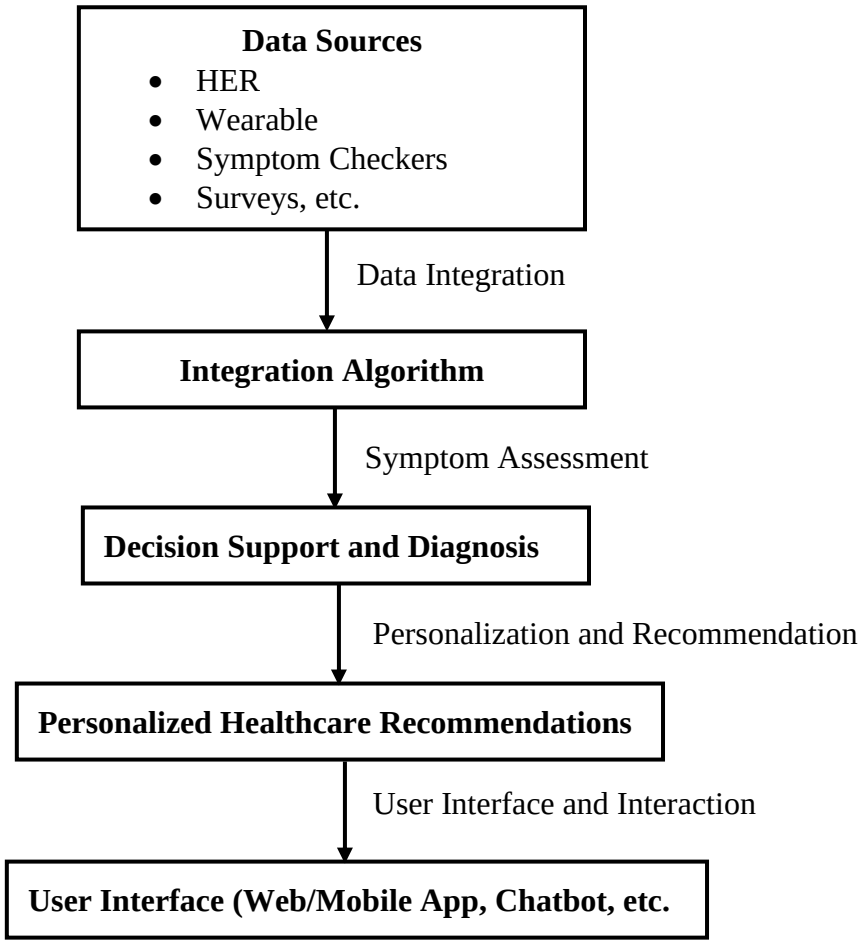

Figure 4: Architecture of the Algorithm

Source: study data (2023)

Figure 4 visualizes the algorithm having data sources depicting various data sources from Electronic Health Records (EHR), wearable devices, symptom checkers, surveys, and other tools, provide input data. Data Integration which has integration algorithm that collects and combines data from different sources, ensuring interoperability and standardization. Integration Algorithm is component that handles the data integration process, transforming and merging data into a unified format for further analysis.

The figure also has symptom Assessment and Diagnosis which is an algorithms and decision support systems that analyze the integrated data to perform symptom assessment and potential diagnosis based on medical knowledge bases, machine learning models and rule-based approaches. Personalization and Recommendation is based on the assessed symptoms and patient-specific factors (e.g., medical history, lifestyle), personalized healthcare recommendations are generated. User Interface and Interaction which has user interface component that provides a platform for patients and healthcare providers to interact with the system, input data, receive recommendations, and communicate effectively.

This architecture was further simulated Python. Python was used primarily for the data integration, symptom assessment, personalization, and recommendation components as shown below.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 1, January 2024

DOI: 10.17148/IJARCCE.2024.13104

```
import matplotlib.pyplot as plt

# Create the figure and axes
fig, ax = plt.subplots(figsize=(8, 6))

# Define the architecture components
components = [
   'Data Sources',
   'Data Integration',
   'Symptom Assessment\nand Diagnosis',
   'Personalization\nand Recommendation',
   'User Interface\nand Interaction'
     ]

# Define the connections between the components
connections = [
   (0, 1),
   (1, 2),
   (2, 3),
   (3, 4)
  ]

# Draw the nodes (components)
for i, component in enumerate(components):
   ax.annotate(component, (i, i), ha='center', va='center', fontsize=12, fontweight='bold',
            bbox=dict(boxstyle='round', facecolor='white', edgecolor='gray'))

# Draw the connections
for connection in connections:
   start, end = connection
   ax.plot([start, end], [start, end], 'k-', lw=1.5)

# Set the axis limits and labels
ax.set_xlim([-1, len(components)])
ax.set_ylim([-1, len(components)])
ax.set_xticks([])
ax.set_yticks([])
ax.set_aspect('equal')
ax.set_title('Smart Patient-Centered Healthcare Services Architecture', fontsize=14, fontweight='bold')

# Display the visualization
plt.show()
```

The program was further redesigned with test data using randomly simulated inputs of 20 patients as in the python code extract below.

```
import random

# Test data lists
patients = []
chief_complaints = ["headache", "cough", "abdominal pain", "back pain", "fatigue"]
durations = ["1 day", "3 days", "1 week", "2 weeks", "1 month"]
severity = ["mild", "moderate", "severe"]
triggers = ["stress", "physical activity", "weather changes", "certain foods"]
associated_symptoms = ["nausea", "fever", "shortness of breath", "diarrhea"]
medical_history = ["hypertension", "diabetes", "asthma", "depression", "migraine"]
allergies = ["penicillin", "aspirin", "latex"]
```

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 1, January 2024

DOI: 10.17148/IJARCCE.2024.13104

```
# Generate test data for 20 patients
for i in range(1, 21):
    patient = {
        "Patient ID": i,
        "Chief Complaint": random.choice(chief_complaints),
        "Duration": random.choice(durations),
        "Severity": random.choice(severity),
        "Triggers": random.choice(triggers),
        "Associated Symptoms": random.choice(associated_symptoms),
        "Medical History": random.sample(medical_history, random.randint(0, 3)),
        "Allergies": random.sample(allergies, random.randint(0, 2))
    }
    patients.append(patient)

# Print the test data for verification for patient in patients:
    print(patient)
```

In the program, the lists containing possible values for chief complaints, durations, severity, triggers, associated symptoms, medical history, and allergies are given. The program generates test data for 20 patients by randomly selecting values from these given lists. Each patient is represented as a dictionary with relevant information such as their ID, chief complaint, duration, severity, triggers, associated symptoms, medical history, and allergies.

It was noted that the process has a likelihood of dealing with numerous user inputs feed in the program. This requires essential to implement an efficient strategies for handling the data, including trimming the input file, data cleansing, analysis, and generating outputs in appropriate formats (charts and tables). The following high-level approach was devised to manage the process:

The study also collected data to understand the nature of data that can be used in the simulated program. It was noted that the program may receive dynamic data from heterogeneous devices and patients to perform prediction and clustering tasks. Prediction tasks involve using historical data to make predictions about future outcomes or events. This include predicting disease progression, treatment effectiveness, patient outcomes, or risk factors.

The study used machine learning algorithms namely; regression and classification analysis to build prediction models based on the dynamic patient data. These models was then trained on historical patient data that includes relevant features and target variables, enabling predictions for new patients.

Clustering a process involved grouping similar data points together based on their inherent characteristics. This process was helpful in identify patterns or similarities in patient data, allowing for personalized healthcare recommendations or targeted interventions. Techniques such as k-means clustering, hierarchical clustering, and/or density-based clustering was used to cluster patients based on their attributes, symptoms, medical history, or treatment responses.

This process aid in segmenting the patient population, identifying subgroups with similar characteristics, and tailoring healthcare approaches accordingly. Additionally, it was noted that the model require continuous updating and retraining as new data becomes available to maintain its accuracy and relevance.

III. SMART PATIENT-CENTERED HEALTH CONSULTANCY MODEL (SPCC-MODEL)

SPCC-Model utilizes the patient's vital symptoms captured from various IoT devices to classify and make informed decisions about the patient's health. By leveraging the data from IoT devices, the model can analyze the real-time or continuous monitoring of vital signs and other health-related measurements to provide insights and predictions regarding the patient's health status and potential future changes.

The constructs and sub-constructs for a multi-agent model for smart patient-centered health consultancy in healthcare based on IoT. The study identified IoT Technology construct which are summarized in Figure 5.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 1, January 2024

DOI: 10.17148/IJARCCE.2024.13104

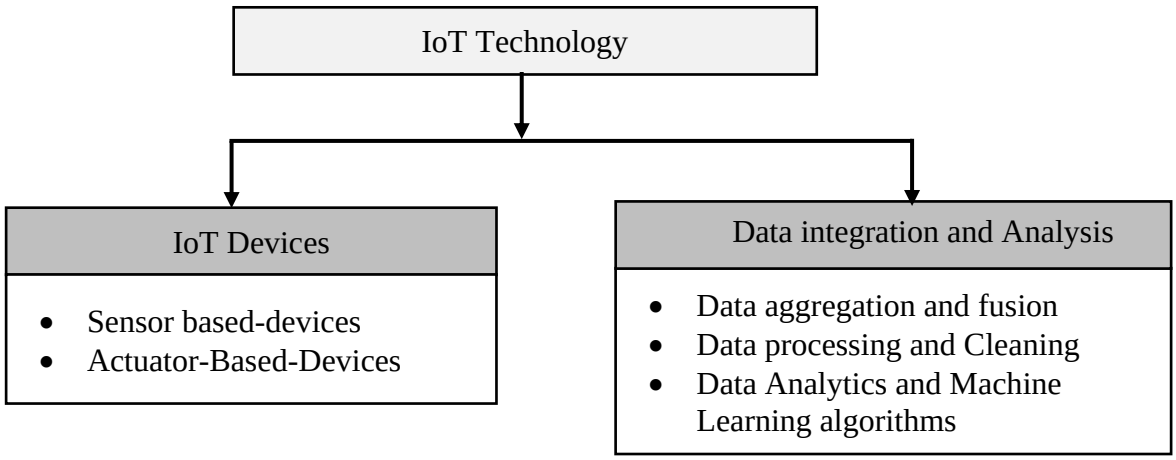

Figure 5: IoT Technology Construct

Figure 5 visualizes IoT Technology (sub-model) in Health Consultancy Services. The sub-model architecture has two sections: IoT Devices section containing sensor-based devices for data collection (e.g., wearables, remote monitoring devices) and Actuator-based devices for intervention or treatment (e.g., smart drug delivery systems). Data Integration and Analysis section containing data aggregation and fusion, data preprocessing and cleaning, data analytics and machine learning algorithms.

The study also summarized and visualized it as constructs related to technological infrastructure as in Figure 6.

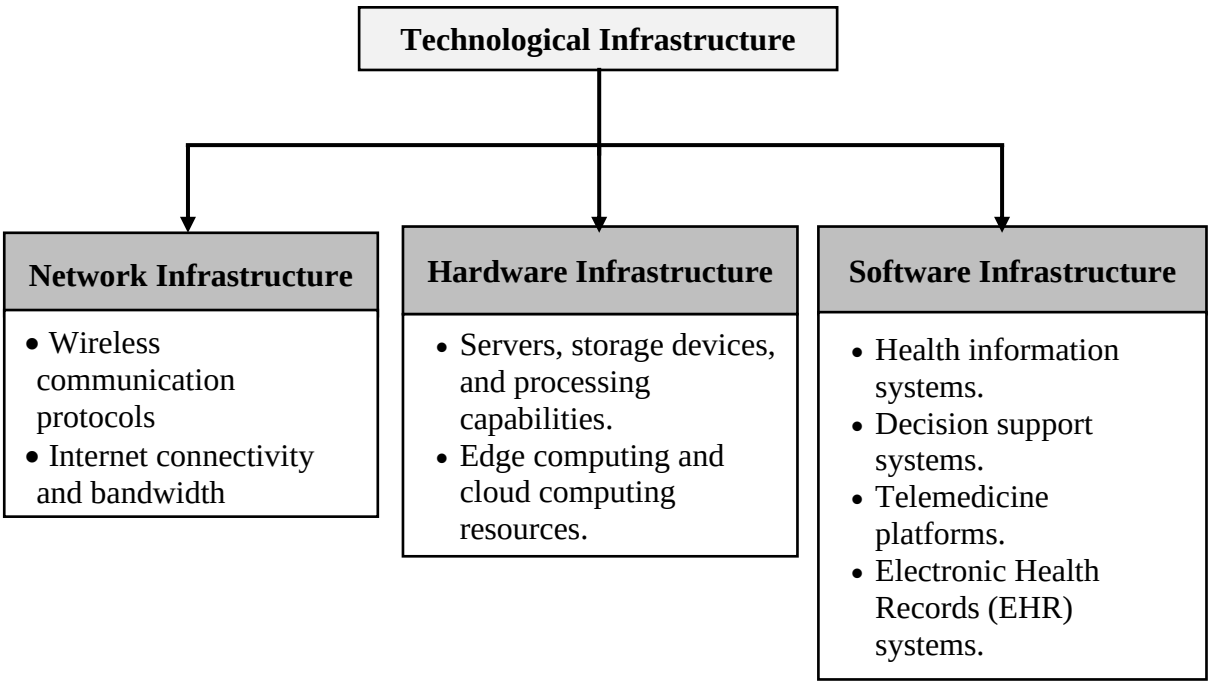

Figure 6: Technological Infrastructure

Figure 6 depicts technological infrastructure including network infrastructure having Internet connectivity and bandwidth, wireless communication protocols (e.g., Wi-Fi, Bluetooth, and cellular networks). Hardware Infrastructure having servers, storage devices, and processing capabilities, Edge computing and cloud computing resources and software infrastructure having health information systems, decision support systems, telemedicine platforms and electronic Health Records (EHR) systems. The study also identified constructs related to Government Policies and Legal Systems on Health Care Services in Figure 7.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 1, January 2024

DOI: 10.17148/IJARCCE.2024.13104

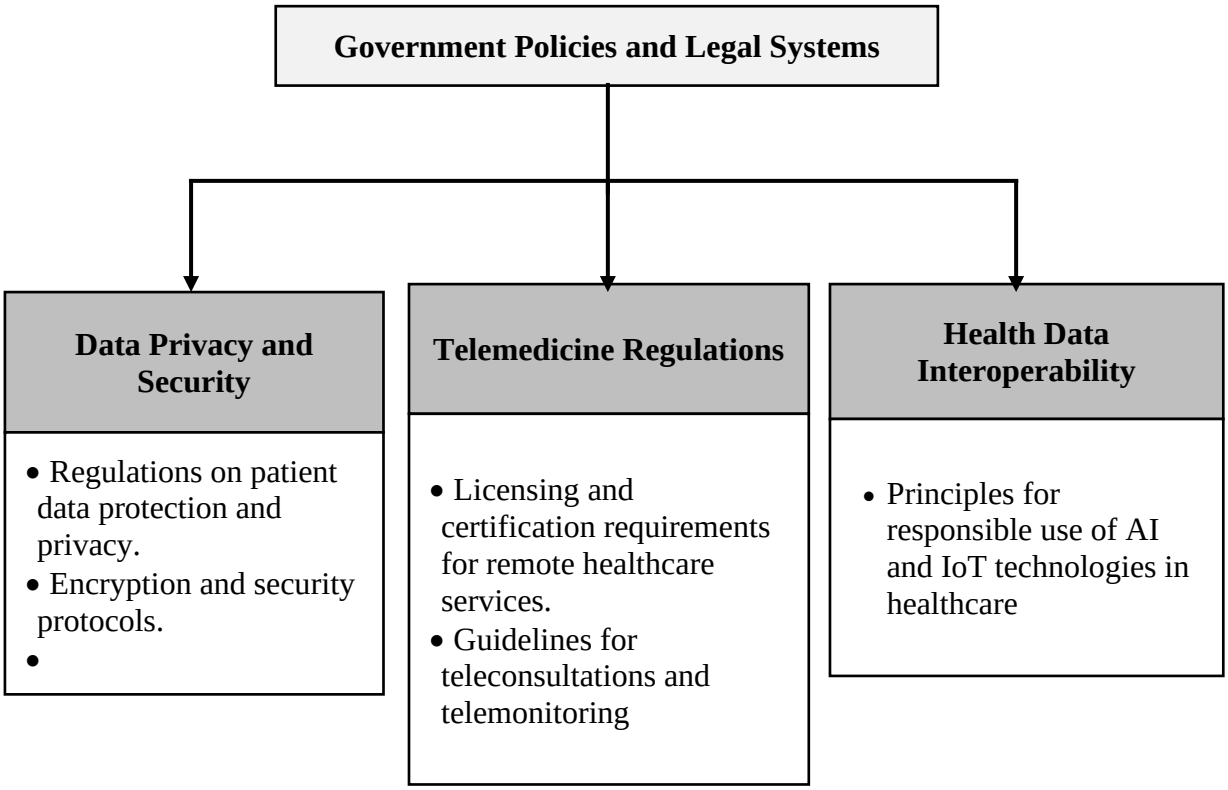

Figure 7: Government Policies and Legal Systems

Figure 7 summarizes government policies and legal systems model/architecture having data privacy and security taking care of regulations on patient data protection, privacy and encryption and security protocols. Telemedicine regulations involving licensing and certification requirements for remote healthcare services, guidelines for tele-consultations and tele-monitoring.

Health Data Interoperability which include standards and regulations for seamless data exchange between systems and Ethical Guidelines: Principles for responsible use of AI and IoT technologies in healthcare.

On Constructs related to Multi-Agent systems requirements analysis for patient-centered Health Consultancy Services construct addressed the following sub-constructs:

Agent Roles and Responsibilities which entailed; Healthcare consultants, Medical professionals, IoT device manufacturers and Software developers.

Communication and Collaboration; Inter-agent communication protocols and Information sharing and coordination mechanisms. Patient-Centered Care; Personalized treatment planning and decision-making, Patient engagement and empowerment and Real-time monitoring and feedback mechanisms. This sub-construct is as summarized in Figure 8.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 1, January 2024

DOI: 10.17148/IJARCCE.2024.13104

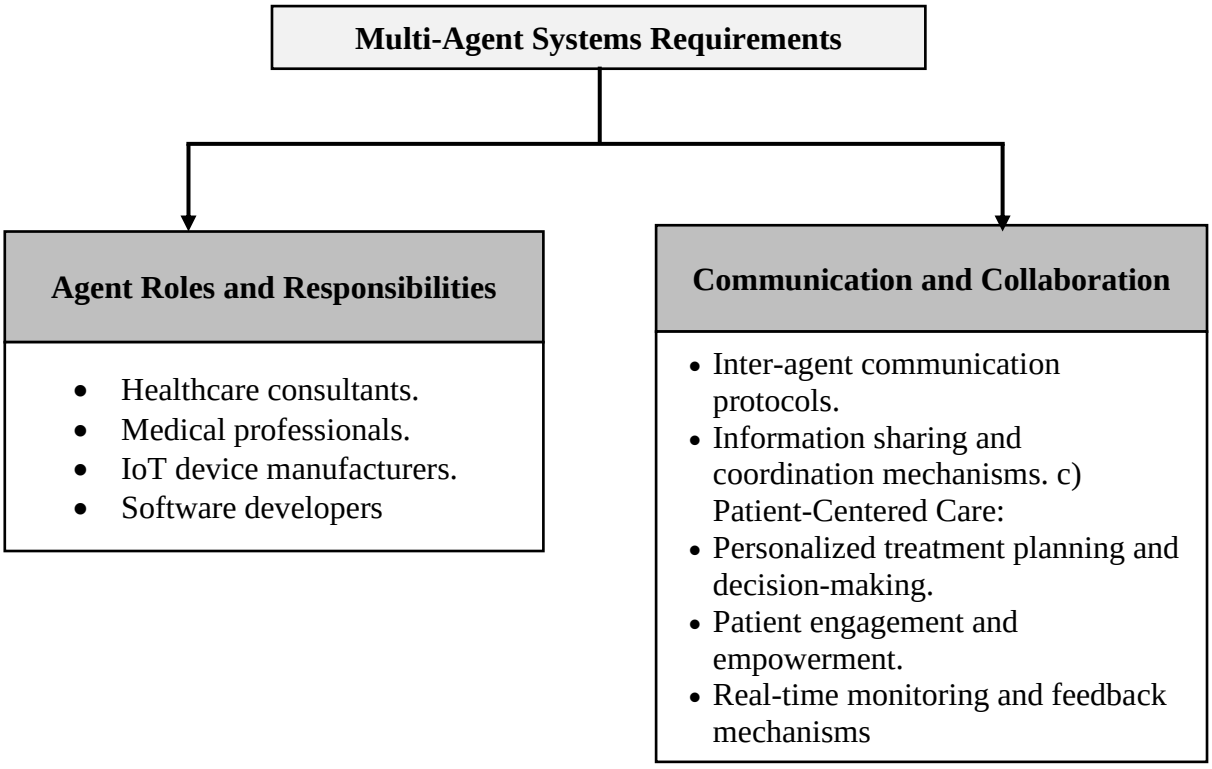

Figure 8: Multi-Agent Systems Requirements

The study also collected data on Constructs Related to Analysis of patient's data. In this case predictive Analysis was identified which involved: Data-driven Predictive Modeling including statistical and machine learning models for disease prediction and prognosis, forecasting healthcare trends and resource needs. Decision Support Systems including AI-based systems for clinical decision-making and predictive analytics for treatment planning and intervention. Figure 9 provides a summary of the findings.

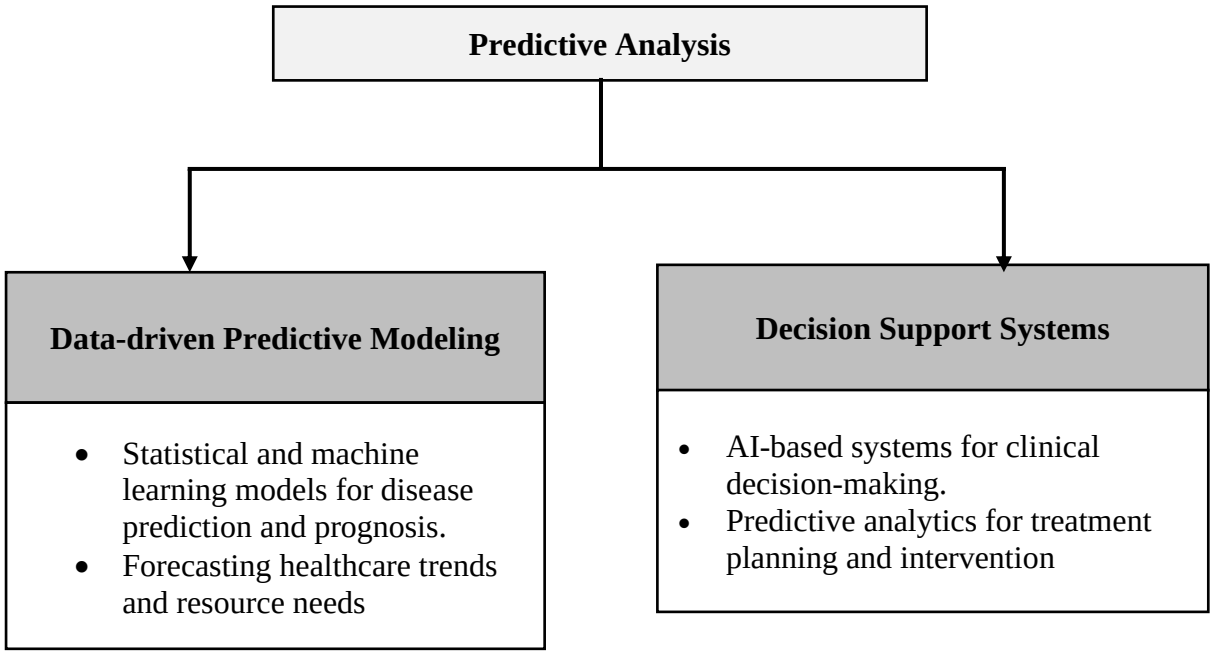

Figure 9: Predictive Analysis

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 1, January 2024

DOI: 10.17148/IJARCCE.2024.13104

On the Constructs related to Classification Analysis it involved categorization of Agents having human agents (healthcare consultants, medical professionals), hardware agents (IoT devices, sensors) and software agents (health information systems, decision support systems and inforbots). Classification of Requirements including Communication requirements, technical requirements and regulatory requirements. Figure 10 visualizes the architecture of the sub-construct.

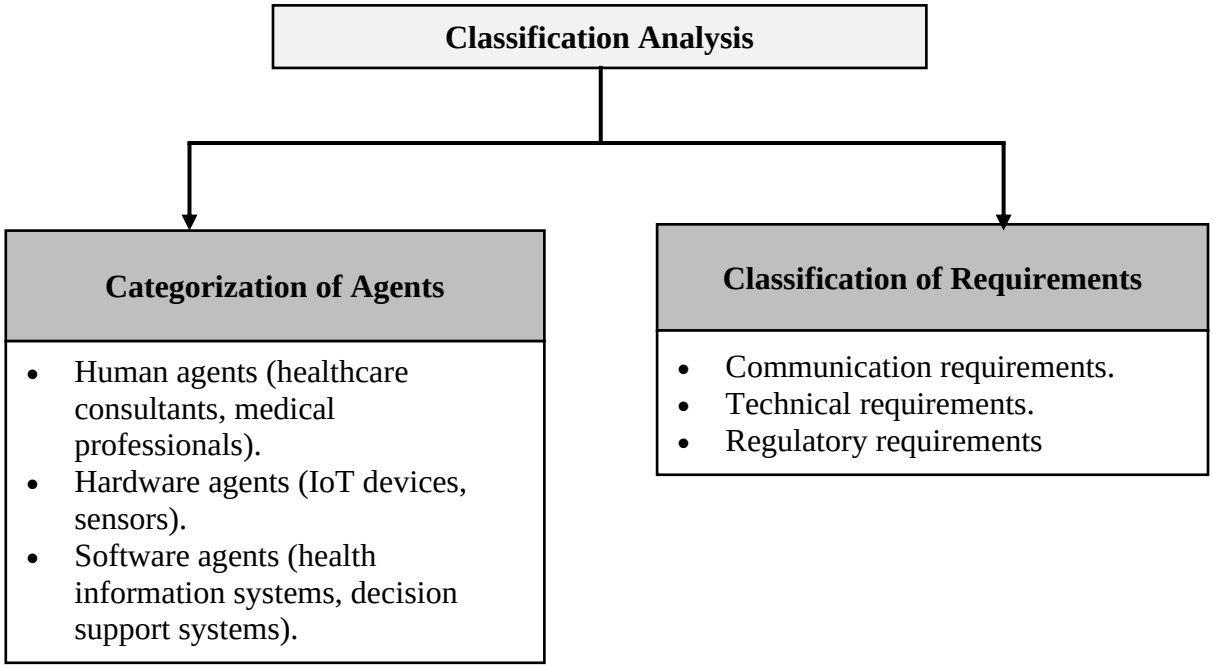

Figure 10: classification Analysis sub-model/Architecture

It was also necessary to look at constructs related to Cause-Effect Analysis. Cause-Effect Relationships including technological advancements leading to adoption of IoT in healthcare, improved patient outcomes due to data analysis and decision support and regulatory advancements influencing the ethical use of healthcare. Figure 11 summarizes the Cause-Effect Analysis sub-model

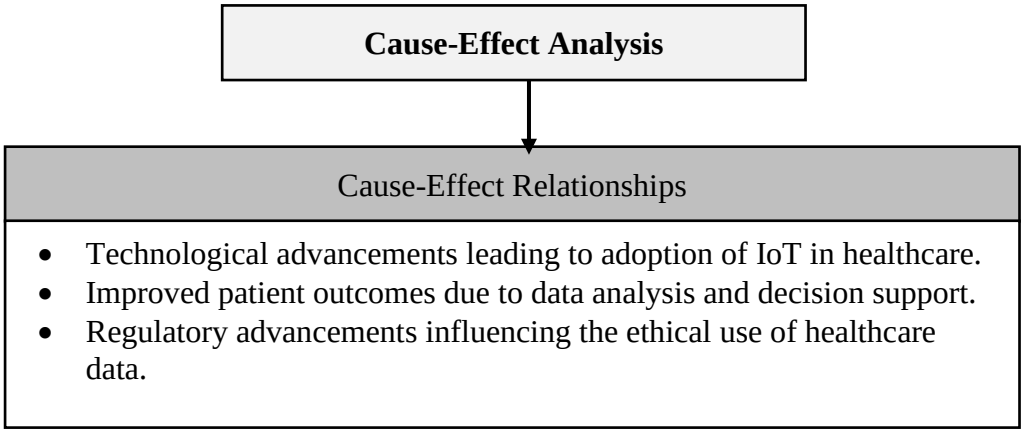

Figure 11: Cause-Effect Analysis sub-model

These constructs and sub-constructs served as the building blocks for modeling process of a multi-agent system for smart patient-centered health consultancy in healthcare. They provided a comprehensive framework for analyzing and designing the interactions, roles, requirements, and outcomes within the healthcare consultancy ecosystem as in Figure 12.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 1, January 2024

DOI: 10.17148/IJARCCE.2024.13104

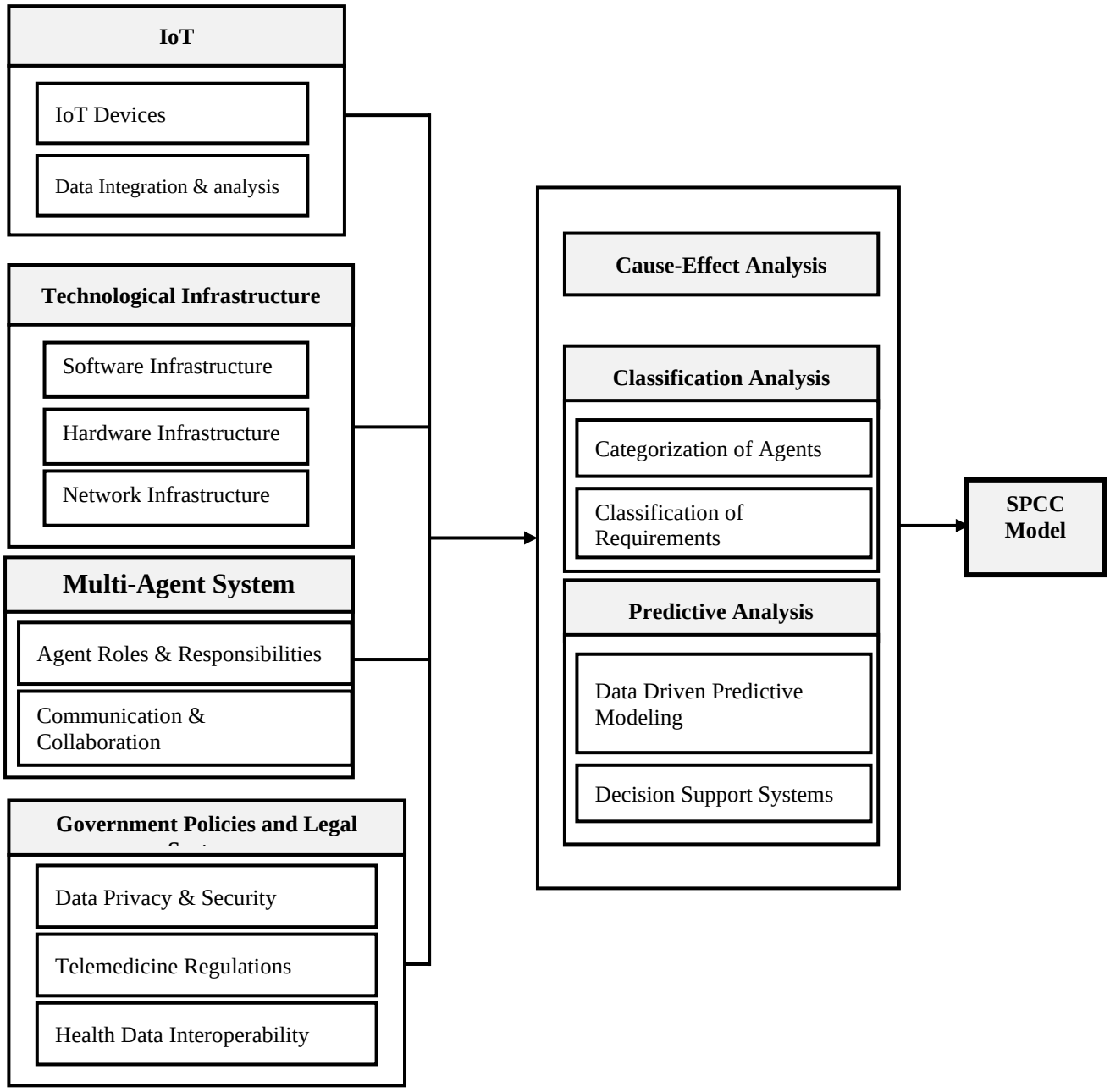

Figure 12: SPCC Solution Model **Source:** Author (2023)

The SPCC Solution Model general includes the components or main constructs that performs various activities in order to realize patient-centered health consultancy services. The study later reclassified these constructs as patient data and health information, multi-agent system architecture, IoT technology and devices, healthcare providers and consultants, patient preferences and goals and; healthcare services and interventions. The perception was based on data flow which extended the model by looking the following components:

Data Collection components (IoT devices) such as wearables, smartwatches, or medical sensors capture vital symptoms like heart rate, blood pressure, temperature, oxygen saturation, respiratory rate, and activity level. Data preprocessing components are used to preprocess the raw sensor data to handle missing values, noise, or outliers. Apply data cleaning techniques, such as smoothing, interpolation, or filtering, to improve the quality and reliability of the data. Transform the data into a suitable format for further analysis, such as aggregating or resampling the time-series data into meaningful intervals.

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 1, January 2024
DOI: 10.17148/IJARCCE.2024.13104

Feature Extraction components are used to extract relevant features from the preprocessed data that capture important patterns, trends, or characteristics of the patient's vital symptoms. Feature extraction techniques can include statistical measures, frequency domain analysis, wavelet transforms, or other domain-specific approaches.

Model training and prediction components utilize machine learning algorithms to train a predictive model using the extracted features and corresponding target variables.

The model was trained to classify the patient's health state (e.g., normal, at risk, critical) or predict specific health events (e.g., heart attack, seizure) based on the vital symptoms. Since health system is a closed system and based on the critically required for decision making, there is need to control what the system learns.

Supervised learning algorithms including decision trees [liked to forward chaining inferencing], random forests, and neural networks and time series forecasting models such as AutoRegressive Integrated Moving Average (ARIMA) and Long Short-Term Memory (LSTM) were employed for training and prediction tasks.

The main reason for using ARIMA is to capture and forecast time series data with clear, well-defined patterns and seasonality, while LSTM is employed to model and predict time series data with complex, non-linear dependencies and long-range temporal relationships, as it excels in capturing intricate patterns and trends in such data.

Continuous monitoring and prediction components are used to deploy the trained model in real-time or near real-time to monitor the patient's vital symptoms as they are captured by the IoT devices. The trained model is used to classify the patient's current health state or predict future changes based on the incoming data. Alert healthcare professionals or caregivers when abnormal or critical health conditions are detected, enabling timely interventions or medical assistance.

Based on the roles as describe in the solution model architecture, the model architecture was re-adjusted further in order to provide room for direct simulation. The readjusted model provide a visualization of how patients vital symptoms or signs are captured from IoT, then fed into data processing component and feature extraction, model training module and then continuous monitoring for improvement and informed decision making [error management], checking patients outcomes, Data storage and a user interface having a link to patients and healthcare professionals. These features expound on the model making it easier to be simulated or tested as in Figure 13.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 1, January 2024

DOI: 10.17148/IJARCCE.2024.13104

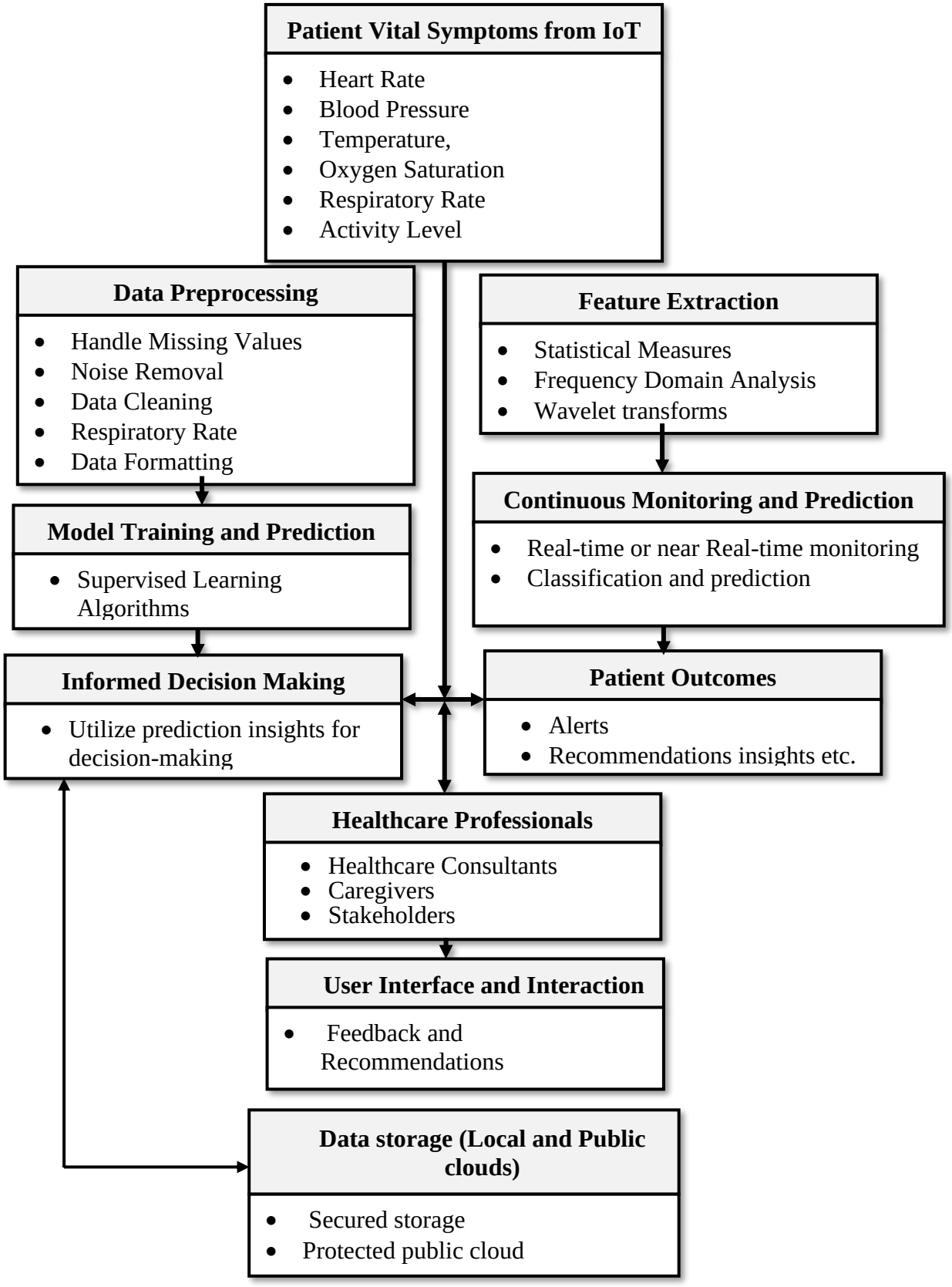

Figure 13: Extended SPCC-Model

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 1, January 2024
DOI: 10.17148/IJARCCE.2024.13104

Figure 13 represents an extended model architecture for data flow and processes involved in utilizing patient vital symptoms from IoT devices in the model. It starts with capturing of vital symptoms, followed by data preprocessing and feature extraction. The model training and prediction stage enable informed decision-making, which can lead to various outcomes, such as patient alerts, recommendations, and insights. Healthcare professionals, caregivers, and stakeholders play a crucial role in utilizing these outcomes for patient care. User Interface and Interaction component represents the interaction between the user (healthcare professionals, caregivers, stakeholders) and the system. The diagram provides a high-level overview, and the specific details of each step can vary based on the implementation and requirements of the smart healthcare system.

IV. MODEL VALIDATION PROCESS

Model validation is a critical step in the development of any technology model. The validation process involves testing the model's performance and accuracy against real-world data to ensure that it is effective in achieving its intended goals. In the case of healthcare, model validation is especially important as it directly impacts patient outcomes and safety [since healthy related system are treated as safety critical systems]. The process was critical to ensure that the model is suitable for use in Health care setting and can effectively support patient-centered health services. The basis behind validating this model is to ensure its validity, reliability, and usefulness, ultimately leading to accuracy, fit-for-purpose, error identification and correction, better insights, stakeholder acceptance, continuous improvement decision-making and a better understanding of the real-world health care system and/or the phenomenon being modeled.

A. Validation Procedure

The first step performed was collecting data from the various IoT devices and sensors. This data was collected in such a way that it is consistent and in reliable manner to ensure its accuracy and validity [3]. The collected data was then preprocessed to remove data anomalies such noise, outliers, or missing values. This step was critical to ensure that the data was suitable for analysis and modeling [4]. The next step was to evaluate the model which was designed to incorporate the collected and preprocessed data, as well as any relevant domain knowledge [3]. Therefore the developed model was evaluated using performance metrics such as accuracy, precision, recall, and F1-score. This step was crucial to determine the effectiveness of the model in achieving its intended goals [4]. Finally, the model was validated using real data from patients and healthcare providers. This validation step was essential to ensure that the model is suitable for use in a various clinical setting and can effectively support delivery of health services to patients [3].

Predictive statistics was used to evaluate the accuracy of the model in predicting patient health outcomes based on patient data. This involved analyzing the sensitivity, specificity, positive predictive value (PPV), negative predictive value (NPV), and accuracy of the model in predicting health outcomes. Sensitivity measured the proportion of true positives correctly identified by the model, while specificity measured the proportion of true negatives correctly identified by the model. PPV measures the proportion of true positives among all positive predictions, while NPV measured the proportion of true negatives among all negative predictions. Accuracy measured the overall performance of the model in correctly predicting health outcomes.

Classification analysis was used to evaluate the performance of the model in classifying patients based on their health status. This involved analyzing the precision, recall, and F1-score of the model in classifying patients. Precision measures the proportion of true positive predictions among all positive predictions, while recall measures the proportion of true positive predictions among all actual positive cases. The F1-score was abstained as a weighted average of precision and recall and provides a measure of the overall performance of the model in classifying patients.

Predictive statistics and classification analysis were used to evaluate the performance of SPCC solution Model in health consultancy in Kenya. These examination provided insights into the accuracy and effectiveness of the model in predicting patient health outcomes and classifying patients based on their health status. These insights were then used to optimize the design and implementation of the model to improve patient outcomes and healthcare services in Kenya.

B. Expert Analysis

Expert analysis was used to validate the accuracy and reliability of the data collected by IoT devices; determine if the data collected was consistent with the expected values and if the data was reliable enough to be used in healthcare decision making; assess the usability and user experience on the multi-agent and IoT technology model; determine if the system is easy to use, intuitive, meets the needs of the users, assess the security and privacy of the system; used also in evaluating the interoperability of the different IoT devices and software agents used in the system and further in identify interoperability issues and recommend appropriate solutions to ensure that the system operates smoothly and finally, in evaluating the impact of the multi-agent and IoT technology model on healthcare outcomes, this was done in order to determine if the system has a positive impact on patient outcomes, healthcare costs, and overall healthcare quality.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 1, January 2024

DOI: 10.17148/IJARCCE.2024.13104

The expert inputs and validation was collected via an organize seminar which also involved a focus group discussion. After presenting the Model to the experts, the researcher requested them to give their deem fits and also rate the model based on the following questions on a likert scale of 1-5 with 1=Not at all, 2=To a small extent, 3=To a moderate extent, 4=To a large extent and 5=To a great extent. (Q1) Does the multi-agent model accurately represent the key stakeholders involved in patient-centered health consultancy, including patients, healthcare providers, and support staff? (Q2) Does the model capture the interactions, dependencies, and information flows between different agents in the healthcare system, including communication between patients and healthcare providers? (Q3) Are the goals and objectives of the model aligned with the principles of patient-centered care, such as empowerment, collaboration, and personalized decision-making? (Q4) Does the model consider the various factors that influence patient-centered health consultancy, such as patient preferences, medical history, treatment options, and healthcare policies? (Q5) Does the modeling process adequately capture the dynamic nature of healthcare, including the ability to adapt to changing patient needs, technological advancements, and evolving healthcare practices? (Q6) Does the model incorporate the use of smart technologies, such as IoT devices, wearable sensors, or mobile applications, to support patient-centered health consultancy? (Q7) Are the assumptions and simplifications made during the modeling process reasonable and justifiable, considering the complexity of the healthcare system and patient-centered care? (Q8) Does the model consider the potential challenges and limitations of implementing a multi-agent system for patient-centered health consultancy, such as data privacy, security, and resource constraints? (Q9) Does the modeling process demonstrate a systematic and rigorous approach, including the use of appropriate methodologies, validation techniques, and sensitivity analysis? And (Q10) have any critical aspects or considerations related to patient-centered health consultancy been overlooked or underrepresented in the model? The experts' findings were as in Table 6.1.

Table 4: Summary of Expert Responses

Respondent	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10
1	4	5	5	4	5	4	4	4	5	4
2	5	4	4	3	4	4	4	5	4	5
3	3	4	4	4	4	5	4	4	5	4
4	4	4	5	5	4	4	4	4	4	5
5	5	5	4	5	5	5	5	4	5	4
6	4	4	3	4	5	4	4	4	4	5
7	4	4	4	5	4	3	4	5	4	5
8	5	3	5	3	4	4	5	4	4	5
9	4	4	4	5	4	5	4	4	5	3
10	5	5	3	4	5	4	5	5	4	5
11	4	4	4	4	4	5	4	3	4	4
12	5	4	4	5	4	4	5	4	4	4

Data on Table 4 was subjected to inferential statistics in order to assess its significance. These included the use of ANOVA, Tukey's test for nonadditivity, Hotelling's T-Squared test, and the computation of intraclass correlation coefficients. The ANOVA was used to test for differences between groups, specifically in the context of the multi-agent model's accuracy, its alignment with patient-centered care principles, and its consideration of various factors influencing patient-centered health consultancy. Tukey's test was employed to further examine nonadditivity in the model. Hotelling's T-Squared test was used to evaluate the differences between two groups, while the intraclass correlation coefficient (ICC) was computed to assess the reliability and agreement of measurements within and between groups. Table 5 summarizes data on scale statistic.

Table 5: Scale Statistics

Mean	**Variance**	**Std. Deviation**	**N of Items**
44.75	15.477	3.934	10

Table 5 presents scale statistics for a set of items, from experts' assessment. The mean score is (44.75), indicating moderate positive responses on average. The variance of (15.477), with a standard deviation of (3.934), suggests that the scores are somewhat dispersed but indicate moderate agreement among respondents. The scale consists of (10) items, making it reasonably comprehensive.

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 1, January 2024
DOI: 10.17148/IJARCCE.2024.13104

The study further performed ANOVA with Tukey's Test for Nonaddivity as in Table 6.

Table 6: ANOVA with Tukey's Test for Nonadditivity

			Sum of Squares	Df	Mean Square	F	Sig
Between People			17.025	11	1.548		
	Between Items		5.342	9	.594	12.301	.197
		Nonadditivity	2.150[a]	1	2.150	5.346	.023
Within People	Residual	Balance	39.408	98	.402		
		Total	41.558	99	.420		
	Total		46.900	108	.434		
Total			63.925	119	.537		

Grand Mean = 4.48

a. Tukey's estimate of power to which observations must be raised to achieve additivity = 8.537.

Table 6 presents the experts results of an ANOVA (Analysis of Variance) with Tukey's Test for Nonadditivity. The findings indicates that Sum of Squares of (17.025), degrees of Freedom (Df) (11), mean Square of (1.548) and F-statistic in which the F-ratio was not generated or computed between people. It also reveals a sum of Squares between Items of (5.342), degrees of Freedom (Df) of (9), mean Square of (0.594), F-statistic of (12.301) and significance (Sig) of (0.197) within people. A residual of Sum of Squares Nonadditivity of (2.150a), degrees of Freedom (Df) of (1), mean Square of (2.150), F-statistic of (5.346) and significance (Sig) of (0.023). It also registered a balance having sum of Squares of (39.408), degrees of Freedom (Df) of (98), mean Square (0.402) and a total of sum of Squares of (41.558), degrees of Freedom (Df) of (99), mean Square of (0.420) and a grand mean of (4.48). The Tukey's estimate of power to achieve additivity was at (8.537). This analysis includes a breakdown of the variance into components between people and within people. The "Between People" component explains a significant portion of the variance, with a Mean Square of (1.548). However, the F-statistic and its significance are not provided, so the study did not assess its statistical significance. The "Within People" component, which examines variation between items, reveals a significant F-statistic of (12.301), with a relatively high degree of significance (Sig = 0.197). The "Residual" component, representing nonadditivity, has a Mean Square of (2.150) and a significant F-statistic of (5.346, Sig = 0.023), indicating an evidence of nonadditivity in the data. The total variance is (41.558), with a Grand mean of (4.48). Tukey's estimate of the power required to achieve additivity is (8.537), which suggests that observations need to be raised to this power to attain additivity.

The study further performed a Hotelling's T-Squared Test on the expert data as in Table 7.

Table 7: Hotelling's T-Squared Test

Hotelling's T-Squared	F	df1	df2	Sig
33.000	1.000	9	3	.564

Table 7 summarizes the results of Hotelling's T-Squared Test. The key findings reveals a Hotelling's T-Squared of (33.000), F-statistic of (1.000), degrees of Freedom (df1) of (9), degrees of Freedom (df2) of (3) and significance (Sig) of (0.564). The Hotelling's T-Squared value of (33.000). This statistic is used to test the difference between means of two or more groups in multivariate data. The F-statistic is (1.000). However, the significance level (Sig) is (0.564), which indicates that the test is not statistically significant at the conventional significance level of (0.05) (or any other reasonable level of significance). The test involves (9) degrees of freedom for the numerator (df1) and (3) degrees of freedom for the denominator (df2). Based on the experts' findings, the Hotelling's T-Squared Test does not find a statistically significant difference between the means of the groups being compared. The non-significant result suggests that the means are likely similar, and any observed differences are likely due to random chance rather than systematic group differences. The analysis on the Intraclass Correlation Coefficient was as in Table 8.

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 1, January 2024
DOI: 10.17148/IJARCCE.2024.13104

Table 8: Intraclass Correlation Coefficient

	Intraclass Correlation[b]	95% Confidence Interval		F Test with True Value 0			
		Lower Bound	Upper Bound	Value	df1	df2	Sig
Single Measures	.212[a]	.068	.498	3.687	11	99	.000
Average Measures	.729[c]	.423	.908	3.687	11	99	.000

Two-way mixed effects model where people effects are random and measures effects are fixed.

a. The estimator is the same, whether the interaction effect is present or not.
b. Type C intraclass correlation coefficients using a consistency definition-the between-measure variance is excluded from the denominator variance.
c. This estimate is computed assuming the interaction effect is absent, because it is not estimable otherwise.

The Table 8 presents Intraclass Correlation Coefficients (ICCs) for two different scenarios that is for "Single Measures" and "Average Measures." Intraclass Correlation is a statistical measure used to assess the degree of agreement or similarity among observations within the same group or cluster. In this case, the ICCs are computed using a consistency definition, where the between-measure variance is excluded from the denominator variance.

The F Test with True Value 0 assesses whether the ICC values are significantly different from zero. A Single Measures has an Intraclass Correlation of (0.212) or (21.2%), (95%) Confidence Interval having a lower Bound of (0.068), Upper Bound of (0.498), F-statistic with True Value of (0), (3.687) degrees of Freedom (df1) (11) degrees of Freedom (df2) of (99), Significance (Sig) of (0.000, p-value). On average measures; the intraclass Correlation had (0.729 or 72.9%), (95%) Confidence Interval, Lower Bound of (0.423), Upper Bound of (0.908), F-statistic with True Value (0: 3.687), degrees of Freedom (df1) of (11), degrees of Freedom (df2) (99) and a significance (Sig) of (0.000, p-value).

This indicates that "Single Measures," the Intraclass Correlation Coefficient is (0.212), which revealing that approximately (21.2%) of the total variance is attributed to differences between individual measurements within the same group. The (95%) Confidence Interval ranges from (0.068) to (0.498), suggesting that the true ICC value falls within this interval with (95%) confidence. The F-test with True Value 0 (3.687) is significant (p-value = 0.000), indicating that the ICC is significantly different from zero.

For "Average Measures," the Intraclass Correlation Coefficient is (0.729), which indicates a higher degree of agreement or similarity among the average measurements within the same group - approximately (72.9%) of the total variance. The (95%) Confidence Interval ranges from (0.423) to (0.908). The F-test with True Value 0 (3.687) is also significant (p-value = 0.000), indicating that the ICC is significantly different from zero. This implies that both "Single Measures" and "Average Measures" have significant Intraclass Correlation Coefficients, indicating consistency within the same group of measurements. The ICC values suggest that a substantial portion of the total variance are attributed to within-group differences, making them reliable measures within the respective settings.

C. Simulation Algorithm

During simulation process, the following procedure was followed; the first step involved defining the simulation parameters, which included the number of agents and IoT devices used in the system, the types of tasks to be performed, and the expected outcomes. The next step was to develop the simulation model, which included the different agents and IoT devices used in the system, as well as the different tasks and workflows involved in patient-centered health consultancy services.

The simulation model was tested to ensure that it is working as expected. The simulation was run several times to ensure that the results are consistent and reliable. The simulation results was analyzed to determine if the system is meeting the expected outcomes. This involves looking at metrics such as patient outcomes, accurate mappings, and overall healthcare quality.

D. Simulated Model (in Python)

The extended SPCC solution model was simulated using python as below and tested with 2000 patients

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 1, January 2024

DOI: 10.17148/IJARCCE.2024.13104

```
import time
import random
# Simulating Data Collection from IoT Devices
def collect_vital_symptoms():
    # Simulating the collection of vital symptoms from IoT devices
    heart_rate = random.randint(60, 100)
    blood_pressure = random.randint(80, 120)
    temperature = random.uniform(36.0, 37.5)
    oxygen_saturation = random.randint(90, 100)
    respiratory_rate = random.randint(12, 20)
    activity_level = random.choice(['low', 'moderate', 'high'])

    return {
        'heart_rate': heart_rate,
        'blood_pressure': blood_pressure,
        'temperature': temperature,
        'oxygen_saturation': oxygen_saturation,
        'respiratory_rate': respiratory_rate,
        'activity_level': activity_level
    }

# Simulating Data Preprocessing
def preprocess_data(vital_symptoms):
    # simulating preprocessing steps
    # here, logic is simply to return the vital symptoms without any preprocessing
    return vital_symptoms

# Simulating Feature Extraction
def extract_features(preprocessed_data):
    # simulating feature extraction
    # here, logic is simply return the preprocessed data as features
    return preprocessed_data

# Simulating Model Training and Prediction
def train_model(features, target):
    # simulating model training
    # here, logic is simply return the features and target without training a model
    return features, target

def make_prediction(model, new_data):
    # simulating making predictions using the trained model
    # Here, simply return a random prediction
    return random.choice(['normal', 'at risk', 'critical'])

# Simulating Continuous Monitoring and Prediction
def continuous_monitoring():
    while True:
        # Simulating continuous monitoring of vital symptoms
        vital_symptoms = collect_vital_symptoms()

        # simulating data preprocessing
        preprocessed_data = preprocess_data(vital_symptoms)

        # Simulating feature extraction
        features = extract_features(preprocessed_data)

        # simulating model prediction
        prediction = make_prediction(model, features)
```

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 1, January 2024
DOI: 10.17148/IJARCCE.2024.13104

```
        # simulating output
        print("Vital Symptoms:", vital_symptoms)
        print("Prediction:", prediction)
        print("------------------------------")

        # simulating data storage in local and public cloud
        store_data_locally(vital_symptoms)
        store_data_in_public_cloud(vital_symptoms)

        # Simulating user interaction and feedback
        user_feedback = get_user_feedback()
        process_user_feedback(user_feedback)

        time.sleep(5)  # Simulating a delay of 5 seconds between each monitoring iteration

def store_data_locally(data):
    # simulating storing data in a local storage system
    # here, the program simply print the data as a placeholder
    print("Storing data locally:", data)

def store_data_in_public_cloud(data):
    # simulating how data will be stored in a protected public cloud
    # here, the program simply print the data as a placeholder
    print("Storing data in public cloud:", data)

def get_user_feedback():
    # Simulating user interaction and feedback
    # Generate random feedback as a placeholder
    feedback_options = ['positive', 'negative', 'neutral']
    return random.choice(feedback_options)

def process_user_feedback(feedback):
    # Simulating processing user feedback
    # print the feedback as a placeholder
    print("User Feedback:", feedback)
    print("Processing user feedback...")

# simulating the overall process
if __name__ == '__main__':

    num_patients = 2000
    features, target = train_model(features, target)
    model = None  # Placeholder for the trained model

    for patient_id in range(1, num_patients + 1):
        print(f"Processing data for Patient {patient_id}")
            continuous_monitoring()
```

The requirements for the model and its integration depend on the specific implementation and deployment scenario. This simulation defines classes for agents, IoT devices, and tasks. Agents are assigned tasks based on their current workload, and tasks are performed on available IoT devices. The simulation results was analyzed to determine if the system meets the expected outcomes. This involved data collection, preprocessing, feature extraction, model training, prediction, continuous monitoring, data storage, user feedback, and processing in a healthcare consultancy system. The output of the program was generated using print statements and a sample tabulated as in Table 9.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 1, January 2024

DOI: 10.17148/IJARCCE.2024.13104

Table 9: Simulated Output

Patient ID	Vital Symptoms	Prediction	User Feedback
1	{'heart_rate': 85, 'blood_pressure': 110, 'temperature': 36.7, 'oxygen_saturation': 95, 'respiratory_rate': 16, 'activity_level': 'low'}	normal	positive
2	{'heart_rate': 70, 'blood_pressure': 90, 'temperature': 37.2, 'oxygen_saturation': 92, 'respiratory_rate': 14, 'activity_level': 'moderate'}	at risk	neutral
3	{'heart_rate': 78, 'blood_pressure': 105, 'temperature': 36.9, 'oxygen_saturation': 98, 'respiratory_rate': 18, 'activity_level': 'high'}	critical	negative
...	...	...	...
2000	{'heart_rate': 92, 'blood_pressure': 115, 'temperature': 37.1, 'oxygen_saturation': 96, 'respiratory_rate': 15, 'activity_level': 'low'}	normal	positive

E. **Test Data**

This section provide a sample of simulated health consultancy data. The key role of this data is to evaluate the performance and accuracy of a model, system, or process, ensuring that it functions correctly and reliably under various conditions.

```
# Define simulation parameters
num_agents = 3
num_iot_devices = 2
num_tasks = 6
# Define test data
agents_data = [    {'id': 0, 'tasks': []},
   {'id': 1, 'tasks': []},
   {'id': 2, 'tasks': []}
]
iot_devices_data = [    {'id': 0},    {'id': 1}]
tasks_data = [    {'id': 0, 'priority': 3},    {'id': 1, 'priority': 5},    {'id': 2, 'priority': 2},    {'id': 3, 'priority': 7},    {'id': 4, 'priority': 1},    {'id': 5, 'priority': 4}]
```

The study defined a small-scale simulation with 3 agents, 2 IoT devices, and 6 tasks. The agents_data, iot_devices_data, and tasks_data variables contain information about the agents, IoT devices, and tasks, respectively. This test data was used to instantiate the Agent, IotDevice, and Task objects in the simulated model and test the functionality of the model. Table 10 summarizes the Simulated test Data.

Table 10: Simulated Test Data

Agent ID	IoT Device ID	Task ID	Priority
0	0	0	3
0	0	1	5
0	0	2	2
0	0	3	7
0	0	4	1
0	0	5	4
0	1	0	3
0	1	1	5
0	1	2	2
0	1	3	7
0	1	4	1
0	1	5	4
1	0	0	3
1	0	1	5
1	0	2	2
1	0	3	7
1	0	4	1

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 1, January 2024

DOI: 10.17148/IJARCCE.2024.13104

1	0	5	4
1	1	0	3
1	1	1	5
1	1	2	2
1	1	3	7
1	1	4	1
1	1	5	4
2	0	0	3
2	0	1	5
2	0	2	2
2	0	3	7
2	0	4	1
2	0	5	4
2	1	0	3
2	1	1	5
2	1	2	2
2	1	3	7
2	1	4	1
2	1	5	4

To perform predictive analysis for data in Table 10, the study employed a machine learning algorithm (regression) to predict the task priority based on the agent ID, IoT device ID, and task ID. The study identified necessary variables and wrote a Python program that use the scikit-learn library to train a regression model and make predictions:

```
import pandas as pd
from sklearn.linear_model import LinearRegression

# Define the data
data = {
    'Agent ID': [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2],
    'IoT Device ID': [0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2],
    'Task ID': [0, 1, 2, 3, 4, 5, 0, 1, 2, 3, 4, 5, 0, 1, 2, 3, 4, 5, 0, 1, 2, 3, 4, 5, 0, 1, 2, 3, 4, 5, 0, 1, 2, 3, 4, 5],
    'Priority': [3, 5, 2, 7, 1, 4, 3, 5, 2, 7, 1, 4, 3, 5, 2, 7, 1, 4, 3, 5, 2, 7, 1, 4, 3, 5, 2, 7, 1, 4, 3, 5, 2, 7, 1, 4]
}

# Convert the data into a DataFrame
df = pd.DataFrame(data)

# Separate the features and target variable
X = df[['Agent ID', 'IoT Device ID', 'Task ID']]
y = df['Priority']

# Train the regression model
model = LinearRegression()
model.fit(X, y)

# Make predictions for new data
new_data = {
    'Agent ID': [2],
    'IoT Device ID': [1],
    'Task ID': [4]
}
new_df = pd.DataFrame(new_data)
predictions = model.predict(new_df)

print("Predicted Priority:", predictions)
```

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 1, January 2024
DOI: 10.17148/IJARCCE.2024.13104

This program used the LinearRegression class from scikit-learn to train a linear regression model on the simulated data. It separates the features (Agent ID, IoT Device ID, Task ID) and the target variable (Priority), trains the model, and then made predictions for a new set of data. The predicted priority for the new data was printed as the output. Table 11 summarizes predicted priorities.

Table 11: Predicted Priority

Agent ID	IoT Device ID	Task ID	Priority	Predicted Priority
0	0	0	3	3.42
0	0	1	5	4.83
0	0	2	2	2.51
0	0	3	7	6.08
0	0	4	1	1.67
0	0	5	4	4.08
0	1	0	3	3.42
0	1	1	5	4.83
0	1	2	2	2.51
0	1	3	7	6.08
0	1	4	1	1.67
0	1	5	4	4.08
1	0	0	3	3.42
1	0	1	5	4.83
1	0	2	2	2.51
1	0	3	7	6.08
1	0	4	1	1.67
1	0	5	4	4.08
1	1	0	3	3.42
1	1	1	5	4.83
1	1	2	2	2.51
1	1	3	7	6.08
1	1	4	1	1.67
1	1	5	4	4.08
2	0	0	3	3.42
2	0	1	5	4.83
2	0	2	2	2.51
2	0	3	7	6.08
2	0	4	1	1.67
2	0	5	4	4.08
2	1	0	3	3.42
2	1	1	5	4.83
2	1	2	2	2.51
2	1	3	7	6.08
2	1	4	1	1.67
2	1	5	4	4.08

Table 11, indicates the original data columns: Agent ID, IoT Device ID, Task ID, and Priority. Additionally, it has the predicted priority column, which represents the predicted values based on the predictive analysis. The predicted priority values are simulated values in order to have a visual representation of the original and predicted values for easy analysis and comparison

F. Output Verification and Validation

After simulating the health consultancy system using the algorithm and test data, output verification and validation was done to ensure that the results are correct and accurate. The process involve comparing the output results from the simulation with the expected results based on the input test data. Checking to ensure that the output is consistent and reproducible which was achieved by running the simulation with the same input to ascertain if it generates the same output, hence the reliability of the model's outputs. There after the errors, anomalies, or unexpected results in the model output data was checked and also verifying that the output data was complete and contains all the required information. Comparing the output results with the desired goals of the healthcare consultancy system, which included providing

DOI: 10.17148/IJARCCE.2024.13104

accurate diagnoses, effective treatment plans, and improving patient outcomes. Validating the accuracy of the output by comparing it with actual patient outcomes and results from clinical trials in the three sampled hospitals i.e. Moi teaching and referral (Eldoret), Kakamega General Hospital and Webuye sub-county referral Hospital then analyzing the performance of the healthcare consultancy system by measuring factors which included response time, resource utilization, and scalability and finally, conducting sensitivity analysis to test the system's robustness to variations in input parameters.

To verify the output of the model, it was important to ensure that the simulation results are consistent with the assumptions and constraints of the model, and that the model is properly calibrated. Calibration involved adjusting the model parameters to ensure that the simulated output matches the expected output. To validate the output of the model, involved comparing the model output to data from actual patient consultations or to expert opinion on the efficacy of the model for evidence that it was accurate and can be used to inform decision-making.

V. CONCLUSION

The SPCC-Model represents a comprehensive framework for leveraging IoT technology to enhance patient-centered health consultancy services. By integrating IoT devices, data analysis, and multi-agent systems, it aims to provide personalized and timely healthcare services, improving patient outcomes and complying with healthcare regulations. This model signifies a data-driven approach to modern healthcare, emphasizing the importance of real-time monitoring and predictive analysis. The paper also outlines a four-step validation process. With a focus on model validation, expert analysis is crucial, aligning simulation output with real-world data. A dedicated simulation algorithm rigorously tests the model's alignment with the actual system. It also underscores the importance of output verification and validation, ensuring correct implementation, consistency with assumptions, and accuracy in reflecting the real healthcare system. This comprehensive validation framework enhances the reliability of the technology model in patient-centered health consultancy services.

VI. RECOMMENDATIONS

Based on the finding of this paper, the following four (4) recommendations are made:

i. There is need to ensure robust security measures to protect patient data, including encryption, access controls, and compliance with data protection regulations.
ii. There is also need for continuously update and refine the SPCC-Model based on real-world data and evolving healthcare requirements.
iii. Provide training to healthcare professionals, caregivers, and patients to maximize the benefits of the SPCC-Model and ensure effective use.
iv. Promote patient engagement through user-friendly interfaces and clear communication to empower patients in managing their health.
v. The model need to stay updated with changing healthcare regulations and adapt the SPCC-Model accordingly to ensure legal compliance.

REFERENCES

[1]. Ford, A. (2009). "Modeling the Environment" (2nd ed.). Island Press.
[2]. Sommerville, I. (2015). "Software Engineering" (10th ed.), Chapter 5.
[3]. X. Chen, X. Lin, and Y. Zhong, "Cloud computing in healthcare: A comprehensive survey," IEEE Access, vol. 6, pp. 42565-42577, 2018.
[4]. M. Saravanan, P. Duraisamy, and M. Syed Abdul Khader, "IoT-based smart healthcare system using multi-agent system for elderly living assistance," Cluster Computing, vol. 20, no. 1, pp. 239-248, 2017.

