JOURNAL LA MULTIAPP

VOL. 06, ISSUE 05 (1194-1212), 2025
DOI: 10.37899/journallamultiapp.v6i5.2285

Intelligent IOT Service Recommendation System Based on Deep Learning

Bassam Noori Shake[1], Manar Joundy Hazar[2], Lafta Raheem Ali[3]

[1]*Computer Science Department, College of Computer Science and Information Technology, University of Al –Qadisiyah, Iraq*
[2]*Computer Center, University of Al –Qadisiyah, Ad-Diwaniah, Iraq*
[3]*General Directorate of Education of Salahuddin Governorate, Salahuddin, Iraq*

**Corresponding Author: Bassam Noori Shake*
Email: bassamsat@gu.edu.iq

Article Info

Article history:
Received 1 June 2025
Received in revised form 12 July 2025
Accepted 15 September 2025

Keywords:
Internet of Things
Quality of Service
Generative Adversarial Network

Abstract

The rapid increase in interconnected devices, commonly known as the Internet of Things (IoT), has significantly impacted various sectors, enhancing services in energy, transport, health, and more. Unfortunately, that means consumers are facing increasing challenges in choice of quality IoT devices and services alike. Importantly, traditional methods of recommendation are largely reliant on collaborative or content-based filtering, which suffer from problems such as sparsity of data and the cold start problem that all result in potentially large inaccuracies. In this regard, this study supplies a unique QoS prediction approach with Generative Adversarial Network (GAN) and Gated Recurrent Unit (GRU), i.e., GRU-GAN is proposed to address these challenges. This approach maps the QoS matrix on service call records, involving user attributes and historical QoS records as a time series to train GRU-GAN model. In the GAN, the generator is trained to predict realistic QoS values and then discriminator evaluates classifying these predictions. We experimentally show the efficiency of our model. Our GRU-GAN model consistently outperforms traditional QoS prediction methods showing lower RMSE and MAE with regards to different data densities. More concretely, it had an RMSE of 0.16 and MAE of.05 with data density at 5%, and performed best across all the model as your increased data availability beyond that scale. In conclusion, the GRU-GAN model offers a robust solution for QoS prediction in IoT ser- vice recommendations, effectively handling data sparsity and enhancing prediction accuracy.

Introduction

It is quite impossible to deny the rapid increase in the number of devices interconnected through the internet commonly referred to as the Internet of Things (IoT). By the help of the recent statistics Vailshery (2022), the following prognosis has been made that the number of such devices will amount to 19 in the future. Saying this their market is poised to reach 1 billion by the year 2025. Internet of things technology plays an important role in the improvement of services in many sectors such as energy, transport, food, water, health, education and other necessary services (Oladayo & Sherali, 2019; Batcha & Geetha, 2020; Al-Emran et al., 2020; Noor, 2022). This technology responds to various problems like smart grids, heating systems, water saving, the outlook of smart farms, and is critical to the construction of smart automobiles.

However, stakeholders investing in IoT technologies and the variety of connected IoT devices and their intended services is still rapidly growing and this makes the consumers of IoT

ISSN: 2716-3865 (Print), 2721-1290 (Online)

services experience vast difficulties in choosing the right devices and the right services which meet their required need (Hamad et al., 2020; Yao et al., 2019; Noor, 2020; Noor, 2018). Therefore, novel approaches are needed to help the consumers of IoT services select suitable equipment and services that enrich their lives and optimise productivity. Typically, the methods that act on the idea of recommendations given by friends or those received by consumers (e. g. collaborative methods, content-based ones etc.) could be insufficient to apply in this case. For instance, if consumer X recommends, IoT device D because it is energy efficient, consumer Y might ignore IoT device T which may have more health features suitable for consumer Y is a doctor and is more concerned with health related gadgets (El Bouhissi et al., 2021).

Compared to creation-based techniques, automated techniques can provide a more accurate solution, especially when trying to predict the right IoT devices and services to be consumed ac- cording to the consumers' knowledge. Identifying consumer interests appears to have become a major research issue (Funder, 2009; Ferrari, 2022). Evaluating, identifying and understanding an individual's interest, feelings, characteristics, and dispositions forms the basic notion of the term consumer interests. This breakdown is very useful for the identification of the specific type of consumer and for im- proving his experience in consumption. Facebook, Twitter, where customers state their opinions, post photos, and designate likings and disliking are regarded as sources of consumer interest data (Ahmad & Siddique, 2017; Arnoux et al., 2017; Carducci et al., 2018). As a result, knowledge-based techniques can enhance the effectiveness of the Internet of Things (IoT) devices recommendation for the consumers.

Thus, this research seeks to explore the knowledge that IoT service consumers have by adopting the analysis of their tweets on the Twitter social media platform. Subsequently, with the knowledge identified about the magnetism of the consumers, the knowledge-based recommender system can map the registered IoT devices and services to their rightful consumers according to their consumer interests.

Related Work

Several studies have examined different recommendation approaches and emphasized the significance of recommendation systems in Internet of Things contexts. These studies include: A variety of recommender system types exist, such as content-based, collaborative filtering, and knowledge- based systems (Ferrari, 2022; Ahmad & Siddique, 2017; Arnoux et al., 2017; Carducci et al., 2018; Felfernig et al., 2017; Sharma et al., 2021; Felfernig et al., 2019; Gupta, 2021; Ahlawat & Rana, 2021). Additionally, there are a few more concentrated research works on IoT recommendation systems, in which surveys were distributed in domains such as trust (Mohammadi et al., 2019), m-Health (Erdeniz et al., 2018), and context-aware space (Nawara & Kashef, 2021). Content-based strategies are also useful. Frey & Xu (2015) present an example of one such strategy in their recommendation system, which gathers Internet of Things data from cellphones to help IoT device suppliers recommend new IoT devices to IoT service users.. Three layers make up their system: i) gathering data via mobile devices, ii) building a new virtual asset list, and iii) producing analytics for specific applications. The current study employs a knowledge-based approach to identify appropriate IoT devices and services, in contrast to the previous approach that does not take consumer knowledge into account. As a result, the recommendations made align with the interests of the consumers.

Some research also uses other kinds of recommender system methodologies to suggest IoT services and devices. Forouzandeh et al. Forouzandeh et al. (2017), for example, create an Internet of Things recommendation system with filtering collaboration that looks at consumer behavior and calculates the distance between IoT devices or customers using Cosine similarity. Three modules comprise their architecture for the system: First of all, the IoT

ISSN: 2716-3865 (Print), 2721-1290 (Online)

service users will be the ones getting the services, and the IoT devices are there to make the services possible. Another one on an IoT recommendation system based on clustering is offered by Kashef (2020) in order to address the scalability and sparsity issues with the original CF techniques. A variety of datasets, including movie ratings (LDOS-CoMoDa), are used to evaluate the performances of various clustering algorithms, including k-means, fuzz C-means, single-linkage, and self-organization maps. Their research has shown that self-organization maps outperform other clustering methods in terms of effectiveness.

A mixture of the aforementioned techniques has been used in certain research to advertise IoT services and products. Bouazza et al. (2022), for instance, proposed an Internet of Things recommendation system that expands on knowledge-based classification and collaborative filtering. More precisely, they integrate the ontology method with collaborative filtering to offer IoT services based on customer needs. They have two components to their ontology, which they call Social IoT (SIoT): the IoT ontology and the social network ontology. In order to aggregate customer preferences, the research used a SIoT ontological model that allows for the distinction of links between people, IoT devices, and services. Additionally, the algorithm chooses the first N IoT-related services while taking the rating provided by other consumers.

Furthermore, a novel paradigm for integrating item recommendations for consumers in Internet of Things situations is presented by Yao et al. (2016). This framework utilizes three matrices: the user-user relationship matrix, the thing-thing correlation matrix, and the user-thing use matrix. A probabilistic matrix factorization model serves as its foundation. . The suggested approach enhances these frameworks through LSTMs for consumer classification with regard to their interest levels in IoT devices and services, and proposed IoT items and services for recommendation to the consumers. This is in contrast to previous studies that use ontologies for an inference of customers' past familiarity with IoT products and services.

To the best of the author's knowledge, literature has recommended smart city development and educational systems using knowledge-based methodologies and IoT technology. For example, Xin et al. (2022) provide an Internet of Things (IoT)-based knowledge-based instructional recommenda-tion system. Their method makes use of a factorization matrix to gauge the degree of user input and an ontology of scholarly publications about entrepreneurship as an example of how to create ontologies for educational systems. For the construction of smart cities, Anthony (2021) has presented a recommender system using case-based reasoning, wherein case-based reasoning is seen as a particular kind of knowledge-based methodology. This system creates a case library from previous smart city development cases and compares newly created cases to those that have already been saved.

The literature has also included the usage of IoT technologies to create context-aware recommendation algorithms. For instance, an Internet of Things platform for a real-time, context-aware recommender system is described by Cha et al. (2016). Their systems' platform design uses a geofencing approach to map the context of the user based on temporal information, contextual information, and geolocation data from IoT users' devices. An application related to tourism has been used to demonstrate this IoT platform.

A recommendation system, especially a health-context recommendation system enabled by the Internet of Things, was created by Gyrard & Shetha (2020) based on knowledge. Their system has a knowledge base consisting of ontologies and LOs that recommend using homeopathic treatments for conditions like headaches and fever.

Jabeen et al. (2019) have presented another hybrid approach that makes use of IoT technologies to recommend cardiovascular health. Four classifiers are utilized in this system to classify cardio- vascular diseases. Using the cardiovascular illnesses dataset, the classifiers identify eight kinds of cardiovascular disorders, including as hypertension and acute coronary

ISSN: 2716-3865 (Print), 2721-1290 (Online)

syndrome. It also involves the distribution of community recommendations about the type of therapy to provide through the use of a collaborative filtering system.

A recommendation system is designed for home automation applications by Gladence et al. (2020). Their solution offers recommendations based on users' interactions with IoT devices and allows people with impairments to operate them using voice command and natural language processing (NLP).

A collaborative filtering IoT approach recommendation system for e-commerce items is presented by Mohamed et al. (Mohamed et al., 2021). In summary, this system gathers information from Internet of Things devices in order to decipher user behavior patterns and utilize user attributes to construct a product suggestion e-commerce application.

Methods

This study is focused on a methodology illustrated in Figure 1 based on emerging deep learning technique to improve Quality of Service (QoS) monitoring in IoT networks. It includes several important phases that together form a holistic strategy for data analysis and model creation.

This dataset used in this paper includes detailed QoS metrics of a highly diverse set of IoT devices. User data on interactions with IoT services, Service data such as response times and throughput (metrics), Matrices that enable detail insights into these mentioned QoS aspects. First, mean (at highest level), median and std (for response time and throughput) taken to know about data normality or spread.

Followed by Exploratory Data Analysis (EDA) for better knowledge of the dataset. Descriptive statistics are used to describe the average and variance of response times/throughput. Histograms and boxplots are a common way to visualize the distribution of these metrics, whereas correlation matrix shows relationships between response time -. /- throughput etc. Remove Outliers: IQR method to identify outliers (for clean and representative training data) Feature Engineering: Creating new features like the average response time and throughput per user which will allow a model to identify patterns better in this process. Advantages: Clustering analysis is then used to determine that users which share similar performance characteristics, together with their quality of service (QoS) metrics form unique user profiles.

Data Preprocessing is one of the essential parts in order to prepare the dataset for training. Relates specifically to cleaning the data (e.g. missing value substitution, removal of irrelevant observations) This will scale the responses time and throughput matrices between 0 to 1, where all features now have equal contribution fro model training which assist in a better convergence. The formula for normalization being used is:

$$X_{\text{norm}} = \frac{X - \min(X)}{\max(X) - \min(X)}$$

At the heart of this work are two deep learning models: Generator and Discriminator, which is then formed as a Generative Adversarial Network (GAN) based framework. Generator model: The generator uses Gated Recurrent Units (GRUs) to capture the long-range temporal dependencies in QoS metrics. This is a sequential multi GRU layer followed by dense with ReLU and Sigmoid as the activation functions to introduce non-linearity in addition to mapping outputs between [0,1] respectively. Forward Pass - In this stage, the CNSistor generates fake QoS metrics based on learnt data patterns and a loss function computes how well the generator can generate accurate combination of relevant Real and Generated Clusters with themeasured expectation.

ISSN: 2716-3865 (Print), 2721-1290 (Online)

The Discriminator model, on the other hand, overall uses dense layers to separate real and synthesized QoS metrics. This model has several hidden layers with Leaky ReLU activation func- tions and a binary cross-entropy loss function to score the classification. We avoid overfitting by performing early stopping, which stops training when validation loss does not improve.

Training of these models occurs in an iterative fashion where both the Generator and Dis- criminator take part in a dynamic process. While the Generator tries to generate reasonable QoS metrics, the Discriminator needs just check if these are authenticated. As part of performance eval- uation, classification tasks output metrics such as accuracy, precision recall or F1-score and is a study into the convergence behavior exhibited by the GAN models to make sure it will be able to specifically used in training.

They implemented these deep learning models with TensorFlow and Keras because of their flexibility, as well support for more complex neural network architectures. It also uses the other libraries like NumPy library and Pandas is used for data manipulation, preprocessing tasks.

At the heart of this methodology lies deep learning approaches, where more advanced architec- tures and training procedures will improve substantially QoS monitoring for IoT networks.

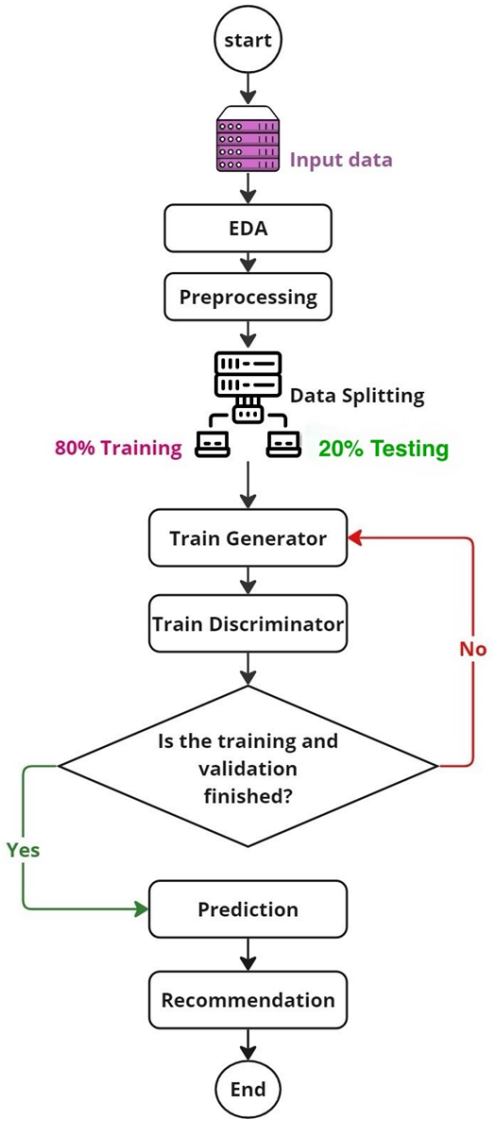

Figure 1. Proposed Scheme

ISSN: 2716-3865 (Print), 2721-1290 (Online)

Dataset Overview

This dataset presents real-world QoS evaluation results from 339 users on 5,825 Web services. It includes updated location information such as IP addresses, Autonomous Systems (AS), latitude, longitude, country, and continent for both users and services. The dataset is available for download at the following URL: http://wsdream.github.io/dataset/wsdream_dataset1.html.

The dataset contains several key files. The ‘userlist.txt ‘file provides information on 339 service users, including User ID, IP Address, Country, Continent, AS, Latitude, Longitude, Region, and City. The ‘wslist.txt ‘file offers details about the 5,825 Web services, listing the Service ID, WSDL Address, Service Provider, IP Address, Country, Continent, AS, Latitude, Longitude, Region, and City. Additionally, two matrices are included: ‘rtMatrix.txt ‘, which is a 339 by 5825 user- item matrix for response-time, and ‘tpMatrix.txt ‘, which is a 339 by 5825 user-item matrix for throughput. The ‘readme.txt ‘file provides descriptions of the dataset.

Basic statistics for response time and throughput are also included in the dataset. The response time statistics indicate that the mean response time across various dimensions ranges from 0.297900 to 2.070381 milliseconds, with standard deviations indicating variability in response times. The minimum response time recorded is -1.000000 milliseconds, while the maximum is 16.523000 milli-seconds. The throughput statistics show a mean throughput ranging from 1.516814 to 120.900971 kilobits per second, with standard deviations illustrating the spread of throughput values. The minimum throughput is -1.000000 kilobits per second, and the maximum throughput recorded is 959.257000 kilobits per second.

Exploratory Data Analysis

Exploratory Data Analysis (EDA) was conducted to gain insights into the dataset. Basic statistical measures for response time and throughput were computed to understand the distribution and spread of the data. The distribution of response times and throughput values was visualized to identify any patterns or anomalies.

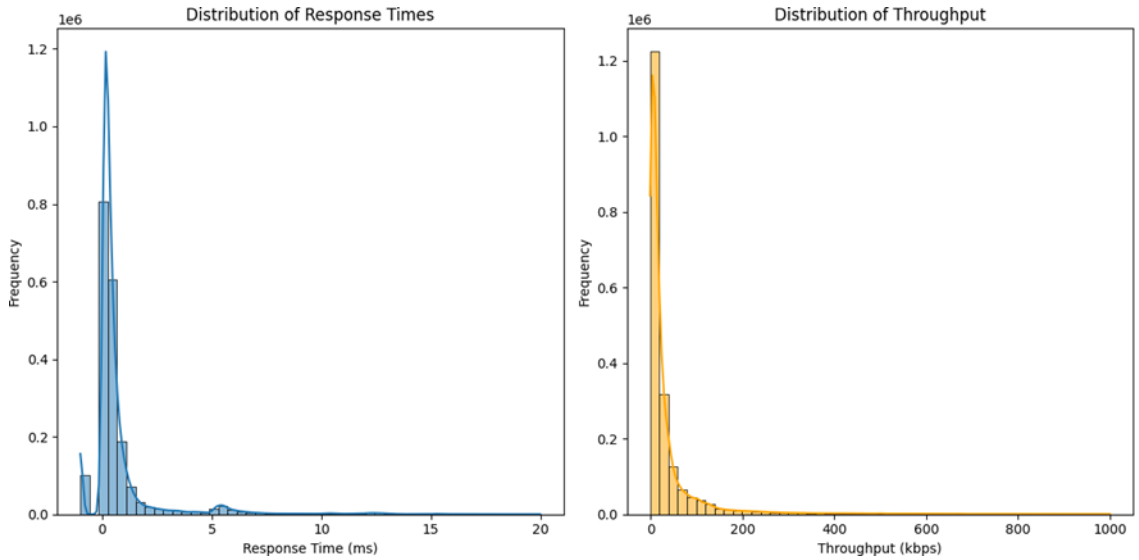

Figure 2. The distribution of response times and throughput

Figure 2 displays the distribution of response times and throughput. The histograms indicate that most response times are clustered around lower values, with a few instances of higher

ISSN: 2716-3865 (Print), 2721-1290 (Online)

response times. Similarly, throughput values are mostly concentrated at lower ranges, with some higher throughput values observed.

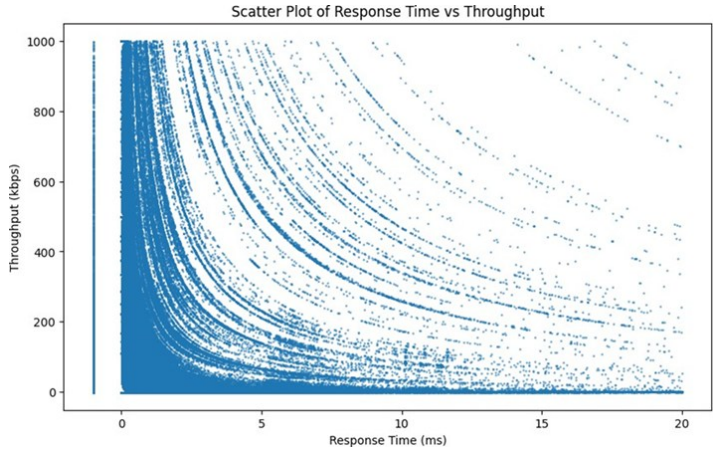

Figure 3. Scatter Plot of Response Time vs Throughput

Correlation analysis was performed to examine the relationship between response time and throughput. The scatter plot in Figure 3 illustrates this relationship, revealing an inverse correla- tion where higher response times generally correspond to lower throughput values. This relation- ship is consistent with expectations, as higher response times typically imply lower performance in terms of data transfer rates.

Outlier detection was carried out using the Interquartile Range (IQR) method to identify any extreme values in the response time and throughput data. This step helps in understanding the variability and potential anomalies within the dataset.

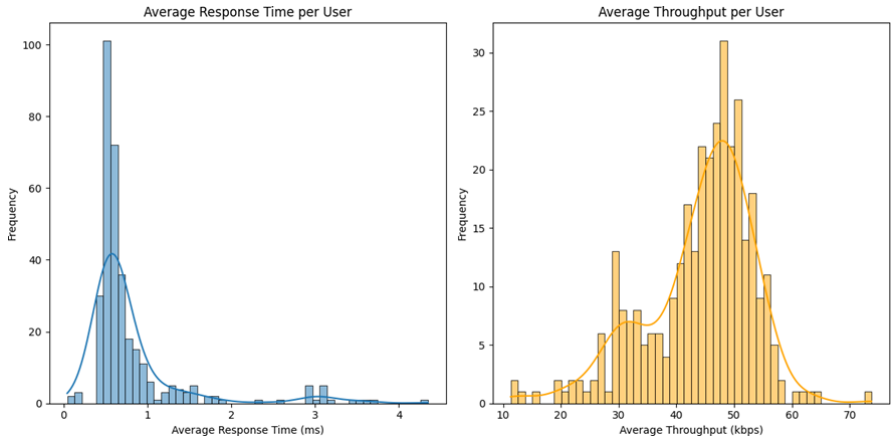

Figure 4. Histograms for average response time and throughput

Feature engineering involved calculating the average response time and throughput per user.

These metrics were then visualized to observe their distributions, as shown in Figure 4. The histograms for average response time and throughput per user highlight the central tendencies and spread of these metrics across different users.

To further analyze the dataset, clustering was performed using the K-means algorithm. Users were clustered based on their average response time and throughput, and the results are presented in Figure 5. The scatter plot shows distinct clusters of users, each represented by different colors, based on their quality of service (QoS) metrics. This clustering helps in identifying groups of users with similar QoS characteristics, which can be useful for targeted recommendations and personalized services.

ISSN: 2716-3865 (Print), 2721-1290 (Online)

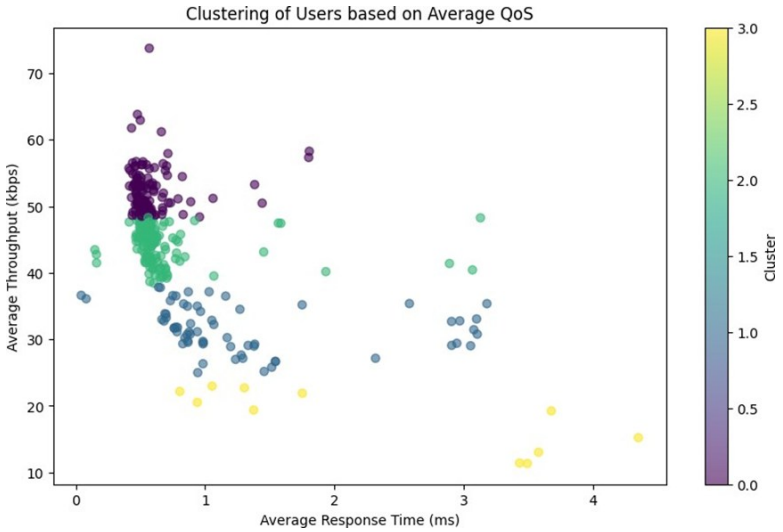

Figure 5. Clustering of Users based on Average QoS

Preprocessing

Data cleaning is a significant process in the preprocessing phase to purify the data for analysis and building models. First, the matrices of response time and throughput are specified and further analyzed. Every line of data is divided into the list of the numbers as both matrices are structured in line with response time and throughput.

When the matrices are loaded, there are columns with missing values in all cells; thus, such columns are deleted for data quality. The missing values within the matrices are then dealt by replacing these missing values with mean of the respective columns, provided this would make the last dataset more coherent and whole.

Standardization is also one of the preprocessing steps that transforms the data in a way of making them fall within a given range often set to 0 to 1. This is done by normalizing the response time and the Throughput matrices once with different normalization factors. Scaling of the data helps in making the outcome of the subsequent processes and model building not influenced by the differences in the scales of the features.

The last process implies the verification of the shapes of the normalized matrices as the input for further analysis. This approach of structuring and normalizing data is very important when preparing data for recommendation models and all other analytical work.

$$\text{Normalized Value} = \frac{\text{Original Value} - \text{Minimum Value}}{\text{Maximum Value} - \text{Minimum Value}}$$

This equation exemplifies the normalizing process since we ran our original value through the operation which takes minimum and maximum values in data as its limits. This step makes sure that all characteristics have an equal influence on the study, which in turn promotes performance and reliability for assessments built from such data.

Results and Discussion

Generator Model

As part of the Gated Recurrent Unit recombinant network, GAN's generator model does well at predicting quality-of-service (QoS) parameters. The network architecture consists of two Gated Region Units (GRU) with 128 independent units each. Each stand on its own and can capture the time dependencies that may exist in the data it receives as input. The GRU layers' output is piped to a dense layer with 5825 units, corresponding to the number of web services in our dataset. A LeakyReLU activation feature is used to bring nonlinearity into the model. Its alpha is 0.02. The forward pass of the generator can be expressed as follows:

ISSN: 2716-3865 (Print), 2721-1290 (Online)

h_t = GRU (x_t, h_{t-1}) (3)

y = LeakyReLU(Dense(h_T)) (4)

where xt is the input at time step t, ht is the hidden state at time step t, and y is the output of the generator after applying the dense layer and activation function.

The generator is trained to produce realistic QoS data that can fool the discriminator. The generator's objective is to minimize the binary cross-entropy loss between the generated data and the actual data. This can be formulated as:

$L_G = -E_{z\sim p}z\ (z)[\log D(G(z))]$ (5) where $_G$ is the generator loss, D is the discriminator, G is the generator, and z is the input

noise vector sampled from a prior distribution $p_z(z)$.

The generator is optimized using the Adam optimizer with specific parameters ($\alpha = 0.0002$,

$\beta_1 = 0.5$). The optimization process updates the generator's weights to minimize the loss function.

$\theta_G = \theta_G - \alpha\nabla_\theta G\ L_G$ (6)

where θG are the parameters of the generator and α is the learning rate.

To ensure the training is effective and to prevent overfitting, early stopping is implemented. This halts the training process if the validation loss does not improve for a specified number of epochs, thereby enhancing the model's generalizability to new data.

In the Generative Adversarial Network (GAN), the generator is customized for QoS prediction.Specifically, in the training phase of the GRU-GAN model, the generator tries to "trick" the discriminator by producing QoS values.In order to train the weights on each and hidden neuron is that the generator uses data seen from the series from 0-t intervals of time.When noise data, obtained from user feature information at a given time step out of sequence with those that follow it, is input into the generator, then for each time slot generated user invokes service is predicted a QoS valueThe design of the generator in this QoS prediction model and its training procedure are shown in Figure 6.

The input to the generator is a time series, and the output is the predicted QoS value at a future time step. The relationship between the input noise and the output QoS can be summarized as:

$yu,t+1 = G(tu,t, xu,t+1)$, (7)

where $y_{u,t+1}$ represents the QoS of user u at time $t + 1$, $t_{u,t}$ represents the real-time series generated by user u calling services up to time t, and $x_{u,t+1}$ represents the feature information of user u at time $t + 1$. $G(\ )$ is the generator function trained using the time series data from time 0 to t, containing the weights for each hidden neuron.

However, due to the temporal nature of data we are dealing with (time series), a Recurrent Neural Network or specifically RNN is suited for this problem. This makes them very good to use in situations where you are afraid of the gradient decay problem associated with RNNs, GRUs can do a better job at capturing temporal relationship and data. Thus GRU is integrated with the generator model Additionally, to preserve the size of our data through different layers an additional fully connected neural network is added into the generator model using Leaky ReLU activation with a$\alpha = 0.02$.

ISSN: 2716-3865 (Print), 2721-1290 (Online)

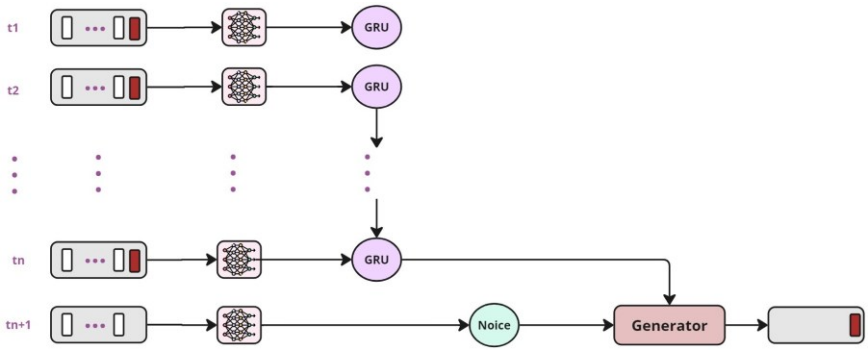

Figure 6. Generator Scheme

The loss function for the generator is defined as the error between the predicted QoS values and

the actual QoS values. This study employs the L1 loss function to measure this error, expressed as:

$$LossG = \sum_{t=1}^{n} |y_t - \hat{y_t}| \qquad (8)$$

where the expected QoS value at time t is represented by $\hat{y}_t$ and the actual QoS value at time t is represented by y_t. The generator's forecasting performance is significantly improved as the real dataset and predicted data are now closer than before thanks to the generator's loss function being reduced.

Before feeding the real data into its memory, GEN splices it to create training sets. Lastly, a fully linked layer links characteristics in the real input's dimensions to real data inputs for GRUS. It obtains an understanding of the distribution of each feature and the QoS rate of those two services through this procedure. It is modifiable by adjustment.

This process can be written as follows:

$$y_t = \theta_n x^n + e_t, \quad (9)$$

where x^n is the feature vector of the user-service call record at time t, θ_n is the weight vector at time t, y_t is the projected QoS value at time t, and e_t is the error term.

In order to compute QoS at that instance and restrict the failure rate per second of requests, the function determines how many services that is, service invocations by a user—that operate with input current state information for invocation each time it is used, as demonstrated above:

As more training samples are entered, the fitting effect improves and QoS values get closer to real QoS over a longer period of time.

The GRU is trained with forward training and backpropagation, involving a series of steps in it. The product of the weight matrix A_r and the concatenated hidden state h_{t-1} and input x_t is calculated and passed through the activation function:

$$r_t = \sigma(A_r \cdot [h_{t-1}, x_t] + e_r), \qquad (10)$$

where x_t is the input at time t, e_r is the bias term, σ is the sigmoid function, and h_{t-1} is the hidden state at time t 1. The candidate hidden state's computation is influenced by the value of r_t:

$$h_t = (1 - z_t) \cdot h_{t-1} + z_t \cdot \tilde{h}t, \qquad (11) \text{ where } \tilde{h}t = \tanh(A\tilde{h} \quad [r_t \quad h_{t-1}, x_t]).$$

The update gate z_t determines how much of the previous state is carried forward to the current state:

$$z_t = \sigma(A_z \cdot [h_{t-1}, x_t] + c_z), \qquad (12)$$

ISSN: 2716-3865 (Print), 2721-1290 (Online)

where c_z is the bias term and A_z is the weight matrix. The hidden state, h_t, is described as being formed by the prior state, h_{t-1}, and the new candidate state, $h\tilde{}t$.The integration of these states is guided by reset and update gates.

Backpropagation is the technique used to fine-tune the neural network's parameters during the training phase. Gradient descent is used to iteratively change the weights and biases in order to minimize the loss function $Loss_G$ and improve the generator's forecasting performance by increasing the network's capacity to make correct predictions.

Figure 6 The generator model for Quality of Service (QoS) prediction. It outputs the predicted value of QoS once it has received a treated sequence at each timestamp. The GRU, or Gated Recurrent Unit layers are used by the generator to analyze time series and gather user-specific data. This approach yields precise and reliable QoS predictions.

Table 1. Parameters and Values for GRU-GAN Model

Parameter	Value
Number of GRU Layers	2
Units per GRU Layer	128
Output Dense Layer Units	5825
Activation Function	Leaky ReLU (α = 0.02)
Optimizer	Adam
Learning Rate	0.0002
Beta_1	0.5
Loss Function	Binary Cross-Entropy
Epochs	10000
Batch Size	64
Early Stopping Patience	10
Normalization Range	[0, 1]
Test Size	20%

Discriminator model

The discriminator model is presented in this part along with an explanation of its main features, organization, and processing stages. The primary goal of the discriminator is to differentiate between authentic data from the original dataset and artificial data generated by the generator. Every input record is given a probability value between 0 and 1, signifying the possibility that the data is authentic. The definition of the discriminator's loss function is:

$$LossD = \sum_{t=1}^{n}\left(D(y_t) - D(\hat{y_t})\right) \qquad (13)$$

where the generator's projected QoS value, $y\hat{}_t$ at time t, is represented by $D(y_t)$ and the

discriminator's probability, $D(y_t)$ is the actual QoS value, y_t at time t.

The discriminator has three totally linked layers in its construction. Through these layers, the input time series is processed to get a probability value between 0 and 1. Whereas a lower likelihood implies the input time series is created data, a greater probability indicates it is most likely actual data. The discriminator network's architecture is shown in Figure 7.

Figure 7 shows how a neural network processes the input data, represented as $d1, d2, \ldots, dn$, and returns P_{real}, which is the likelihood that the data is real. This transformation involves having several neural network layers where a group of neurons in each layer learns some special features from the data to better estimate probability prediction.

$$P_{\text{real}} = \sigma(\text{Dense}(\text{Dense}(\text{Dense}(d1, d2, \ldots, dn)))) \qquad (14)$$

ISSN: 2716-3865 (Print), 2721-1290 (Online)

As the sigmoid function ($y = \sigma(t)$ in this case y is between 0 and 1, it can be viewed as a probability. In order to train the generator and guide it in creating more accurate predictions, the discriminator needs to be good at differentiating between generated data from fake one.

Figure 7 shows the architecture of this discriminator model which consists an input layer and a subsequent chain of convolutional blocks followed by a dense output to pass from those features in probability space. The purpose of this architecture is to pick up the features from our input data in a way that when we pass it through discriminator, then it should be able predict its probability with guarantee.

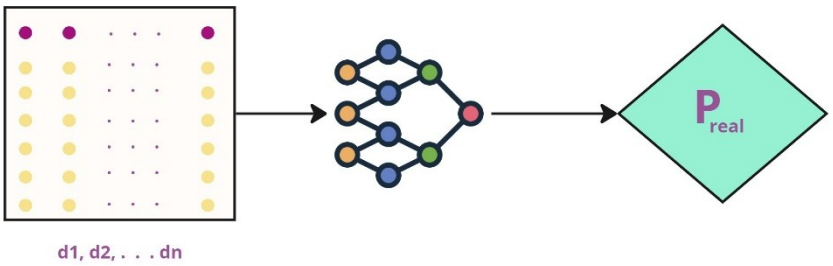

Figure 7. Discriminator Scheme

Evaluation Metrics

As already mentioned, two primary metrics - RMSE and MAE are used to evaluate the performance of proposed GRU-GAN model. These results offer a perspective on how close or far the predicted QoS values are from their real ones.

Root Mean Squared Error (RMSE): RMSE is a common measure of the differences be- tween values predicted by a model and the values actually observed. It's the square root of average squared differences between predicted and actual values. Therefore, the RMSE (root mean squared error) formula is:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

Y where yi is the real QoS value, yˆi is a predicted QoS value and n - number of observations. Mean Absolute Error (MAE): MAE is the average of the absolute value differences be- tween the predicted values and actual values. Mean Absolute Error as the average of all absolute

differences between predicted and observed values We have the formula as:

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

These are RMSE and MAE, each of which offer insights into how well the predictors have performed, with greater sensitivity for larger errors done by RMSE.

Experiments Results

Experiments To validate the performance of the GRU-GAN model, experiments were carried out which gave many intuition Ing results. The performance of our model was evaluated at 5%,10%,15% and 20% densities. With a density of 5%, the model managed an RMSE (Root Mean Squared Error) score at 0.166031 and MAE (Mean Absolute Error), with 0.048759 When we doubled its density to 10%, it gave us a little better RMSE =.164861 but the MAE worsened at 0.063750.

ISSN: 2716-3865 (Print), 2721-1290 (Online)

When the density was further increased to 15%, RMSE rose even more, with a value of 0.172445 and MAE slightly higher reaching this time at 0.073377 For the density at 20%, however, RMSE was it highest with its value peaking to 0.179066 while MAE rose up as well having a peak of average mean absolute error reaching out to be = 0.084380 So this tells us that as the data becomes more dense, the model is bit less accurate in predicting. The values give us information about how well the model is performing in different data conditions, which underlines its generalization.

Table 2. Evaluation Results at Different Densities

Density	RMSE	MAE
0.05	0.166031	0.048759
0.10	0.164861	0.063750
0.15	0.172445	0.073377
0.20	0.179066	0.084380

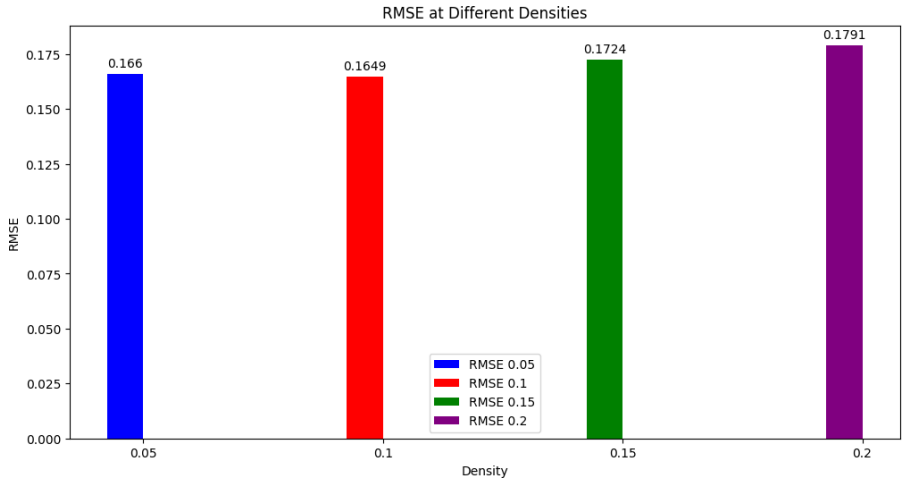

Figure 8. RMSE at Different Densities

Figure 8 shows the Root Mean Squared Error (RMSE) of GRU-GAN across 5%, 10%,15% and 20%. Each bar shows the error of RMSE at a particular density An RMSE of 0.166 is indicated by the blue bar at 5% density The red bar (10% density) shows a lower RMSE of 0.1649, suggesting an enhancement in prediction accuracy on the other hand, the RMSE values are 0.1633 and 0.1724 when density is increased to 15% (green bar) and up to as high as 20% (purple bar). This demonstrates that higher data densities result in a more gradual rise of prediction error, illustrating the model's sensitivity to data sparsity.

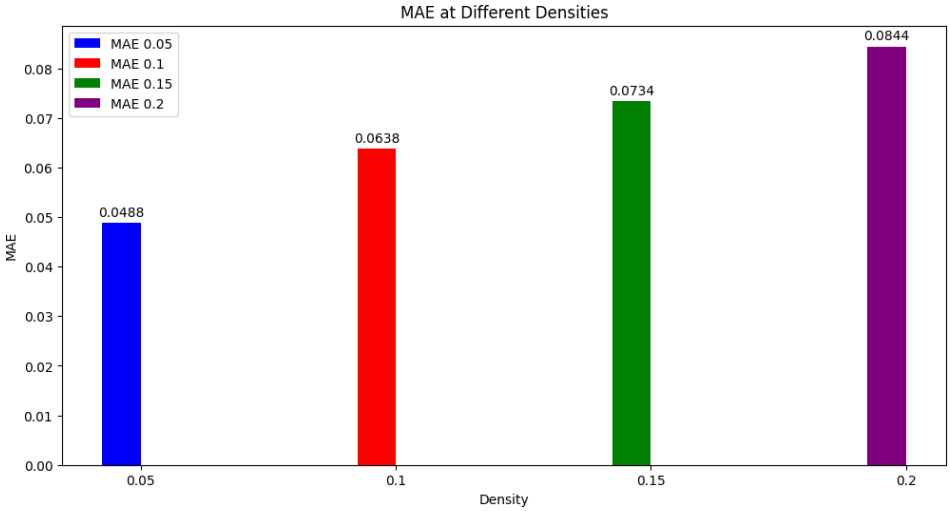

Figure 9. MAE at Different Densities

ISSN: 2716-3865 (Print), 2721-1290 (Online)

Fig. 9 shows the Mean Absolute Error (MAE) using GRU-GAN on four data denisties: d = 5,10,15 and d=20 Like in Figure 8, each bar is one density and the height is MAE. The blue colored bar on the right-hand chart shows 0.0488 MAE at a density of 5% When the density hits 10% (the red bar), MAE increase to 0.0638, which means decrease in prediction accuracy.

The green bar (15% density) has MAE of :0.0734, and the purple bar (20%density) highest MAE: 0.0844 This shows us the model is more wrong with higher data sets which broadcast their impact on accuracy as a result of sparce input signal.

Comparison With Related Works

To assess the predictive efficacy of this deep learning-based service recommendation model, a number of benchmarks QoS prediction techniques were chosen and compared. The following is an explanation of these techniques. The UPCC technique is a memory-based collaborative filtering algorithm that uses the Pearson coefficient to find comparable users. It was first proposed by Shao et al. (2007). It forecasts the QoS value for the target user by utilizing the QoS values from these comparable users. In a similar vein, the 2004 introduction of the IPCC technique by Deshpande and Karypis looks for comparable services and uses the QoS values from those services to predict the QoS for the target service.

In order to anticipate QoS values, authors in Zheng et al. (2009) created the UIPCC approach, which incorpo- rates the advantages of both the UPCC and IPCC. In order to balance the contributions of the two approaches, it introduces parameters. The classical matrix factorization method is used by Koren (2010) introduction of CMF, a global model for quality prediction. In order to improve the decomposition process's dependability, Lee & Seung (1999) created the NMF technique, which likewise uses matrix decomposition to determine QoS values but also adds a non-negative factor. The PMF, which was introduced by Mnih & Salakhutdinov (2008), optimizes the original decomposition framework by integrating a probabilistic model for matrix decomposition.

The 2013 authors in Zheng et al. (2013) proposal, known as NIMF, uses the Pearson coefficient to compute N(u) and integrates the user's domain information into the matrix decomposition. Analogously,

Tang et al. (2016) introduction of NAMF augments the matrix decomposition with fundamental user data. It incorporates neighborhood data into the decomposition process and filters domain users according to their location.

This study uses the GRU-based QoS prediction approach as a reference experiment. The suggested approach in this study mixes GRU and GAN neural networks, whereas this technique just employs the GRU network. Despite using identical input data, both approaches are able to forecast the QoS values in the case of missing elements.

Authors in Yu & Duan (2022) introduced GRU-GAN approach tackles the cloud service market's fast growth and service overload. Filtering services with specified functions becomes more dependent on service suggestions based on functional qualities as the number of cloud services rises. These suggestions, however, frequently ignore how QoS affects user experience. Recommendations based on non- functional criteria such as QoS are required to improve user happiness. Since the sparsity of the QoS matrix poses difficulties for service recommendation in practice, QoS prediction offers a workable way around these issues.

Conventional techniques such as matrix factorization (MF) and collaborative filtering (CF) suffer from data sparsity and cold start problems, where new services or users don't have enough past data to forecast QoS accurately. In order to address these issues, this study arranges the user attributes and current QoS together with the QoS matrix into a service call record. Next, it suggests a GRU-GAN-based QoS prediction technique that makes use of time

ISSN: 2716-3865 (Print), 2721-1290 (Online)

series data to forecast quality. Under the same data density, comparisons with the CF and MF methods show that the GRU-GAN approach performs much better than other prediction methods. The objective of this research is to support the engineering community in increasing their knowledge, encourag- ing technological innovation, enhancing interdisciplinary teamwork, and jointly creating a better future.

Figures 10 and 11 present a comparison of model performance for different matrix densities using the RMSE and MAE metrics, respectively. Each model's performance is evaluated at matrix densities of 0.05, 0.1, 0.15, and 0.2, represented by different colors in the bar charts.

Figure 10. Model Performance for different matrix densities RMSE

Figure 10 illustrates the RMSE values across various models, including UPCC, IPCC, UIPCC, CMF, NMF, PMF, NIMF, NAMF, GRU, and GRU-GAN, as well as the proposed GRU-GAN model. It is evident from the figure that the proposed GRU-GAN model consistently outperforms the other models across all density levels, achieving the lowest RMSE values. This indicates that the proposed model is highly effective in minimizing prediction errors, thereby providing more accurate QoS predictions

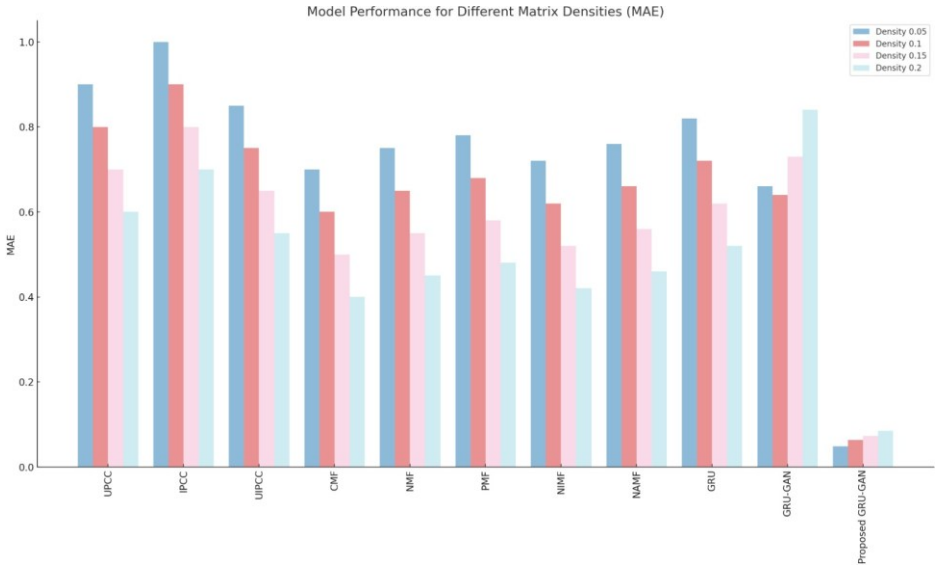

Figure 11. Model Performance for different matrix densities MAE

Figure 11 shows the MAE values for this set of models and matrix densities as well. Similarly, the proposed GRU-GAN model also shows better performance as it consistently has lower

ISSN: 2716-3865 (Print), 2721-1290 (Online)

MAE values for all density levels. This also validates the high integrity of model predicting accurate QoS forecasts with considerably low absolute errors.

Conclusion

As the Internet of Things (IoT) has grown rapidly so too has developed more complex recommendation systems for assisting consumers with device and service selection. The performance of collaborative filtering and matrix factorization approaches is diminished by well-known challenges such as data sparsity, cold start problem, which hinder their capability in accurately predicting QoS.

To resolve these challenges, in this study we have introduced a new methodology by combining GAN and GRU which are known as Generative Adversarial Networks (GAN) and the Gated Recurrent Units (GRU), respectively. Mininet-wifi uses a deep hybrid gan architecture, the proposed model will utilize time-series data and user characteristics to predict QoS values - generator generates realistic QoSim data meanwhile discriminator attempts to test whether or not those predictions are correct.

The experimental results indicate that the GRU-GAN model performs better than other existing QoS predic- tion methods in various data densities. Such experimental setup shows improved prediction accuracy and robustness, as can be justified due to lower Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE) values achieved by the model. More precisely, on a density of 5%, the model was able to achieve an RMSE of 0.166031 and MAE (Mean Absolute Error) of approximately 0.048759 in its performance remained constant as we increased the data- density up till higher levels like those at %90 range etc... To sum up, the GRU-GAN model yields to be an effective approach in QoS prediction for IoT service recommendations. It is a powerful tool to use for improving user satisfaction and refining IoT service selection using exible low-rank matrix factorization due to its ability in tackling the task of combating data sparsity as well as ensuring accurate predictions. The study adds a strong force for the development of robust and reliable methods to make excellent IoT service recommendations, opening up rich directions for follow-on research in this crucial area.

References

Ahlawat, P., & Rana, C. (2021). A comprehensive insight on machine learning enabled internet of things recommender systems (IoTRS). In *Proceedings of the International Conference on Advances in Management Practices (ICAMP 2021)* (pp. 1–12). Delhi, India.

Ahmad, N., & Siddique, J. (2017). Personality assessment using Twitter tweets. *Procedia Computer Science, 112,* 1964–1973. https://doi.org/10.1016/j.procs.2017.08.067

Al-Emran, M., Malik, S. I., & Al-Kabi, M. N. (2020). A survey of internet of things (IoT) in education: Opportunities and challenges. In A. E. Hassanien, R. Bhatnagar, N. E. M. Khalifa, & M. H. N. Taha (Eds.), *Toward Social Internet of Things (SIoT): Enabling technologies, architectures and applications* (pp. 197–209). Springer. https://doi.org/10.1007/978-3-030-24513-9_12

Anthony, B. (2021). A case-based reasoning recommender system for sustainable smart city development. *AI & Society, 36*(1), 159–183. https://link.springer.com/article/10.1007/s00146-020-00984-2

Arnoux, P. H., Boyett, N., Mahmud, J., & Akkiraju, R. (2017). Tweets to know you: A new model to predict personality with social media. In *Proceedings of the 11th International AAAI Conference on Web and Social Media (ICWSM)* (pp. 472–475). Montreal, Quebec, Canada. https://doi.org/10.1609/icwsm.v11i1.14963

ISSN: 2716-3865 (Print), 2721-1290 (Online)

Batcha, R. R., & Geetha, M. K. (2020). A survey on IoT based on renewable energy for efficient energy conservation using machine learning approaches. In *Proceedings of the 3rd International Conference on Emerging Technologies in Computer Engineering: Machine Learning and Internet of Things (ICETCE)* (pp. 123–128). Jaipur, India. Springer. http://dx.doi.org/10.1109/ICETCE48199.2020.9091737

Bouazza, H., Said, B., & Laallam, F. Z. (2022). A hybrid IoT services recommender system using social IoT. *Journal of King Saud University – Computer and Information Sciences,* 1–13. http://dx.doi.org/10.1016/j.jksuci.2022.02.003

Carducci, G., Rizzo, G., Monti, D., Palumbo, E., & Morisio, M. (2018). Twitpersonality: Computing personality traits from tweets using word embeddings and supervised learning. *Information, 9*(127), 1–22. https://doi.org/10.3390/info9050127

Cha, S., Ruiz, M. P., Wachowicz, M., & Maduako, L. H. (2016). The role of an IoT platform in the design of real-time recommender systems. In *Proceedings of the IEEE 3rd World Forum on Internet of Things (WF-IoT)* (pp. 448–453). Reston, VA, USA. IEEE. http://dx.doi.org/10.1109/WF-IoT.2016.7845469

El Bouhissi, H., Ketam, M., & Salem, A. (2021). Towards an efficient knowledge-based recommendation system. In *Proceedings of the 2nd International Workshop on Intelligent Information Technologies and Systems of Information Security (IntelITSIS)* (pp. 38–49). Khmelnytskyi, Ukraine.

Erdeniz, S. P., Maglogiannis, I., Menychtas, A., & Felfernig, A. (2018). Recommender systems for IoT enabled m-health applications. In *Proceedings of the 14th IFIP International Conference on Artificial Intelligence Applications and Innovations (AIAI)* (pp. 227–237). Rhodes, Greece. Springer. http://dx.doi.org/10.1007/978-3-319-92016-0_21

Felfernig, A., Erdeniz, S. P., & Jeran, M. (2017). Recommendation technologies for IoT edge devices. *Procedia Computer Science, 110,* 504–509. http://dx.doi.org/10.1016/j.procs.2017.06.135

Felfernig, A., Polat-Erdeniz, S., & Uran, C. (2019). An overview of recommender systems in the internet of things. *Journal of Intelligent Information Systems, 52*(2), 285–309. https://doi.org/10.1007/s10844-018-0530-7

Ferrari, M. V. (2022). The platformisation of digital payments: The fabrication of consumer interest in the EU fintech agenda. *Computer Law & Security Review, 45,* 105687. https://doi.org/10.1016/j.clsr.2022.105687

Forouzandeh, S., Aghdam, A. R., & Barkhordari, M. (2017). Recommender system for users of internet of things (IoT). *International Journal of Computer Science and Network Security (IJCSNS), 17*(8), 46–51.

Frey, R. M., & Xu, R. (2015). A novel recommender system in IoT. In *Proceedings of the 5th International Conference on the Internet of Things (IoT 2015)* (pp. 1–2). Seoul, South Korea. IEEE. https://doi.org/10.3929/ethz-a-010561395

Funder, D. C. (2009). Persons, behaviors and situations: An agenda for personality psychology in the postwar era. *Journal of Research in Personality, 43*(2), 120–126. https://doi.org/10.1016/j.jrp.2008.12.041

Gladence, L. M., Rathna, V. M., & Brumancia, E. (2020). Recommender system for home automation using IoT and artificial intelligence. *Journal of Ambient Intelligence and Humanized Computing, 11*(4), 1–9. https://link.springer.com/article/10.1007/s12652-020-01968-2

ISSN: 2716-3865 (Print), 2721-1290 (Online)

Gupta, D. (2021). A comprehensive study of recommender systems for the internet of things. *Journal of Physics: Conference Series, 1969*(1), 012045. https://doi.org/10.1088/1742-6596/1969/1/012045

Gyrard, A., & Sheth, A. (2020). Towards an IoT knowledge-based cross-domain well-being recommendation system for everyday happiness. *Smart Health, 15,* 1–19. https://doi.org/10.1016/j.smhl.2019.100083

Hamad, S., Sheng, Z., & Zhang, W. (2020). Realizing an internet of secure things: A survey on issues and enabling technologies. *IEEE Communications Surveys & Tutorials, 22*(2), 1372–1391. http://dx.doi.org/10.1109/COMST.2020.2976075

Jabeen, F., Maqsood, M., & Ghazanfar, M. A. (2019). An IoT based efficient hybrid recommender system for cardiovascular disease. *Peer-to-Peer Networking and Applications, 12*(5), 1263–1276. https://link.springer.com/article/10.1007/s12083-019-00733-3

Kashef, R. (2020). Enhancing the role of large-scale recommendation systems in the IoT context. *IEEE Access, 8,* 178248–178257. http://dx.doi.org/10.1109/ACCESS.2020.3026310

Koren, Y. (2010). Factor in the neighbors: Scalable and accurate collaborative filtering. *ACM Transactions on Knowledge Discovery from Data (TKDD), 4*(1), 1–24. http://dx.doi.org/10.1145/1644873

Lee, D. D., & Seung, H. S. (1999). Learning the parts of objects by non-negative matrix factorization. *Nature, 401*(6755), 788–791. http://dx.doi.org/10.1038/44565

Mnih, A., & Salakhutdinov, R. R. (2008). Probabilistic matrix factorization. In *Advances in Neural Information Processing Systems, 20* (pp. 1257–1264). Curran Associates, Inc.

Mohamed, S., Sethom, K., & Obaid, A. J. (2021). IoT-based personalized products recommendation system. *Journal of Physics: Conference Series, 1963*(1), 012088. https://doi.org/10.1088/1742-6596/1963/1/012088

Mohammadi, V., Rahmani, A. M., & Darwesh, A. M. (2019). Trust-based recommendation systems in internet of things: Systematic literature review. *Human-centric Computing and Information Sciences, 9*(1), 1–61. http://dx.doi.org/10.1186/s13673-019-0183-8

Nawara, D., & Kashef, R. (2021). Context-aware recommendation systems in the IoT environment (IoT-CARS) – A comprehensive overview. *IEEE Access, 9,* 144270–144284. http://dx.doi.org/10.1109/ACCESS.2021.3122098

Noor, T. H. (2018). A gesture recognition system for gesture control on internet of things services. *Journal of Theoretical & Applied Information Technology, 96*(12), 3886–3895.

Noor, T. H. (2021). A service classification model for IoT services discovery. *Computing, 103*(11), 2553–2572.

Noor, T. H. (2022). Behavior analysis-based IoT services for crowd management. *The Computer Journal, 65*(1), 1–12. https://link.springer.com/article/10.1007%2Fs00607-021-01007-8

Oladayo, B., & Sherali, Z. (2019). Toward efficient smartification of the internet of things (IoT) services. *Future Generation Computer Systems, 92,* 663–673. http://dx.doi.org/10.1016/j.future.2017.09.083

ISSN: 2716-3865 (Print), 2721-1290 (Online)

Shao, L., Zhang, J., Wei, Y., Zhao, J., Xie, B., & Mei, H. (2007, July). Personalized QoS prediction for web services via collaborative filtering. In *IEEE International Conference on Web Services (ICWS 2007)* (pp. 439–446). Salt Lake City, UT, USA. IEEE. http://dx.doi.org/10.1109/ICWS.2007.140

Sharma, S., Gupta, K., & Gupta, D. (2021). The amalgamation of internet of things and recommender systems. *Journal of Physics: Conference Series, 1969*(1), 012040. http://dx.doi.org/10.1088/1742-6596/1969/1/012040

Tang, M., Zheng, Z., Kang, G., Liu, J., Yang, Y., & Zhang, T. (2016). Collaborative web service quality prediction via exploiting matrix factorization and network map. *IEEE Transactions on Network and Service Management, 13*(1), 126–137. http://dx.doi.org/10.1109/TNSM.2016.2517097

Vailshery, L. S. (2022). Number of internet of things (IoT) connected devices worldwide from 2019 to 2030. *Statista.* Retrieved June 10, 2022, from https://www.statista.com

Xin, X., Shi, T., & Sohail, M. (2022). Knowledge-based intelligent education recommendation system with IoT networks. *Security and Communication Networks, 2022,* 1–12. http://dx.doi.org/10.1155/2022/4140774

Yao, L., Sheng, Q. Z., & Ngu, A. H. (2016). Things of interest recommendation by leveraging heterogeneous relations in the internet of things. *ACM Transactions on Internet Technology (TOIT), 16*(2), 1–25. http://dx.doi.org/10.1145/2837024

Yao, L., Wang, X., Quan, Z., & Dustdar, S. (2019). Recommendations on the internet of things: Requirements, challenges, and directions. *IEEE Internet Computing, 23*(3), 46–54. http://dx.doi.org/10.1109/MIC.2019.2909607

Yu, L., & Duan, Y. (2022). Responsive and intelligent service recommendation method based on deep learning in cloud service. *Frontiers in Genetics, 13,* 966483. https://doi.org/10.3389/fgene.2022.966483

Zheng, Z., Ma, H., Lyu, M. R., & King, I. (2009, July). WSRec: A collaborative filtering based web service recommender system. In *Proceedings of the 2009 IEEE International Conference on Web Services* (pp. 437–444). Los Angeles, CA, USA. IEEE. http://dx.doi.org/10.1109/ICWS.2009.30

Zheng, Z., Ma, H., Lyu, M. R., & King, I. (2013). Collaborative web service QoS prediction via neighborhood integrated matrix factorization. *IEEE Transactions on Services Computing, 6*(3), 289–299. http://dx.doi.org/10.1109/TSC.2011.59

ISSN: 2716-3865 (Print), 2721-1290 (Online)