https://doi.org/10.38104/vadyba.2026.1.02

Vadyba
Journal of Management
2026, № 1 (42)
ISSN 1648-7974

DEVELOPING A DECISION SUPPORT INTERFACE SOFTWARE FOR PROPERTY VALORIZATION SYSTEM

İlhan Tarımer[1], Simona Grigaliūnienė[2], Zubair Tahir[3]
[1]*Muğla Sıtkı Koçman University, Turkey*
[2]*Lithuania Business College, Lithuania*
[3]*Freelancer Computer Engineer, Pakistan*

Abstract
The real estate industry frequently faces challenges in achieving accurate property valuations, a crucial factor in property transactions. Evaluating commercial real estate tenants is crucial for various stakeholders in the real estate industry, such as property owners, investors, lenders, and property managers. Due to the challenges, this study aims to present a novel Decision Support (DS) software developed to address those challenges by integrating both front-end and back-end systems for seamless, data-driven property assessments. DSS can play a critical role in ensuring accurate property valuations – a key aspect of real estate transactions that often relies on outdated or static data. While the front-end side offers an intuitive interface for users to input property data and receive near-instant results, the back-end side is powered by advanced machine learning algorithms to process this information efficiently. This dual system can ensure that users not only receive accurate valuations but also benefit from a seamless and intuitive user experience. In this paper, we outline the development process, the technologies which we have used, and the evaluation methods which was applied to ensure that the DSS is reliable and effective. The Decision Support System's software for property valorization was developed through comprehensive and multi-stage processes. This system not only enhances the accuracy of valuations but also streamlines the overall decision-making process. The DSS addresses several key challenges in property valuation, including fluctuating market conditions and the complexities of data interpretation. Its scalable design and the continuous learning capabilities of its AI models ensure that the system remains relevant and accurate as market dynamics change. The development of the Decision Support Software (DSS) for property valorization represents a significant step forward in real estate technology. The software's ability to provide accurate, real-time valuations, combined with its user-friendly interface and detailed market insights, positions it as a valuable tool for real estate professionals, investors, and regulators alike.
KEY WORDS: Property valuation, Decision support system (DSS), Machine learning, AI Models, Real estate, Market trends.
JEL classification: R3

Introduction

The DSS software(s) tackles with issues such as fluctuating market conditions and external factors that impact property values. To ensure clarity throughout this paper, the abbreviation DSS is exclusively used to refer to the Decision Support System developed for property valuation. This terminology aligns with its standard usage in decision-making and information systems literature. Any prior reference to "Decision Support Software" or "DS" has been unified under the term DSS for consistency.

By incorporating AI models that continuously learn and adapt, the software improves the accuracy of valuations over time. Furthermore, the decision support system (DSS)'s back-end is built with scalable databases to handle substantial datasets, ensuring its suitability for both small-scale and large-scale property markets. In addition to providing property valuations, a DSS offers actionable insights, making it a comprehensive tool for professionals, investors, and regulators. The software's dual functionality enhances decision-making processes in the real estate market. This paper details the software development process, the materials and methods used, and evaluates its performance, highlighting its potential as a vital tool in modern real estate practices.

Decision Support Systems (DSS) are widely used across various industries to assist decision-makers by analyzing large amounts of data and offering relevant insights. In the real estate sector, DSS can play a critical role in ensuring accurate property valorization – a key aspect of real estate transactions that often relies on outdated or static data. Evaluating commercial real estate tenants is crucial for various stakeholders in the real estate industry, such as property owners, investors, lenders, and property managers (Bytautė, 2024). Traditional valorization methods struggle to account for rapid market changes and the influence of diverse factors such as economic shifts and property characteristics.

In response to these challenges, this study introduces a Decision Support Software (DSS) specifically designed for property valuation. Our DSS integrates both front-end and back-end components to provide users with reliable, real-time valuations. The front-end offers a user-friendly interface that allows easy input of property data and instant access to valorization results. On the back–end, advanced AI models continuously update and refine their predictions based on new data, ensuring that the system remains accurate and reflective of real-time market conditions. This dual system ensures that users not only receive accurate valuations but also benefit from a seamless and intuitive user experience.

DSS which was developed is unique in its ability to combine accurate property valorization with comprehensive market insights, giving users a broader understanding of the real estate landscape. The software is designed to serve real estate professionals, investors, and regulators by providing a flexible, scalable solution that adapts to various market dynamics. Decision support Systems for real estate improve decision-making efficiency in sectors where rapid changes occur. Similarly, Ezenwaka (2025) have argued that AI-powered property valorization systems can revolutionize real estate by providing up-to-date, dynamic assessments. Siddiqui, et. al. (2025) have used comprehensive machine learningbased system for forecasting property prices, highlighting how machine learning enhances predictive accuracy in valorization models. It is explained that decision support systems enhance adaptability in dynamic markets. Yim & Chung (2025) proposed framework that support the responsible and effective integration of AI into property valuation.

In response to these challenges, this study introduces a Decision Support Software (DSS) specifically designed for property valuation. Our DSS integrates both front-end and back-end components to provide users with reliable real-time valuations. The front-end offers a user-friendly interface for easy input of property data, while the back - end utilizes AI models that continuously update and refine their predictions based on market conditions.

Grybauskas, et. al. (2021) demonstrated predictive analytics by a web-scraping algorithm as collecting 18,992 property listings in the city of Vilnius. Al-Rimawi & Nadler (2025) have identified the added value of smart city technologies in real estate development that would transform traditional real estate into smart ones. Root et. al. (2023) developed a data, reasoning, usefulness (DRU) framework for a multi-faceted performance assessment in real estates. McKinsey & Company (2023) have suggested that generative AI techniques can also be integrated to provide more nuanced real estate insights. This dual system ensures that users not only receive accurate valuations but also benefit from a seamless and intuitive user experience.

Our DSS is unique in its ability to combine accurate property valorization with comprehensive market insights, giving users a broader understanding of the real estate landscape. The software is designed to serve real estate professionals, investors, and regulators by providing a flexible, scalable solution that adapts to various market dynamics.

As highlighted by Amannah & Izuchuk (2023), scalable solutions can drive long-term innovation in property management systems. Barlybayev, et. al. (2024) addresses the challenge of accurately determining the fair market value of real estate valorizations in Kazakhistan, ensuring more precise and data-driven decision-making processes. Kumar, et. al. (2019) presents solutions to problems of that designing tools for investment and preserving the privacy of the entire process. Vincenzo et. al. (2019) also discuss the importance of cloud infrastructure for supporting real estate decision-making, noting its critical role in scalability and accessibility. Patel (2023) predicts that the convergence of AI and big data will continue to shape the future of real estate, enabling more sophisticated and data-rich property assessments. The profound impact AI has on property valorization models is explored as highlighting the technological advancements that make these systems more accurate. Kaur & Solomon (2021) stress the aims to examine the extent of property technology, where automation and intelligent systems can optimize property operations and valuations. Liu, et. al. (2025) have found multi-source data fusion and analysis algorithms improving the efficiency of real estate management to emphasize how technological advancements improve the accuracy and efficiency of real estate decision-making.

Belton et al. (2019) discuss the role of AI in smart manufacturing and data privacy, which parallels the growing need for technical standards in real estate technologies. In addition, Che, et al. (2014) explored how intellectual property rights enforcement affects market entry, an issue that is particularly relevant in the development of proprietary real estate technologies. Vallejo-Alonso, et. al. (2015) investigated the financial valorization of intangible assets, a concept that is applicable to the growing field of intellectual property in real estate technology. Finally, Parchomovsky & Siegelman (2002) provide a unified theory of intellectual property, emphasizing the importance of integrating various theoretical perspectives to address contemporary challenges in property valorization and management.

Materials and Methods

The Decision Support System's software for property valorization was developed through comprehensive and multi-stage processes. The tools, methodologies, and technologies used to build and deploy the DSS are described.

Dataset Overview. The Decision Support System (DSS) developed in this study was trained and evaluated using a curated dataset of approximately 16,000 real estate listings collected from Zameen.com, one of Pakistan's leading property platforms. The data was gathered over the course of 2022–2023 and reflects real-time market activity in various urban regions, with a focus on Lahore.

Each listing in the dataset contained key attributes, including the property's location, size (in marlas or square feet), price, number of bedrooms and bathrooms, and property type (e.g., house, plot, apartment). Because the data was sourced from live listings, it offers a genuine representation of actual market dynamics rather than theoretical estimates.

The raw data underwent preprocessing to ensure quality and consistency. This included the removal of duplicate entries, treatment of missing values, standardization of price formats, and the transformation of categorical fields into machine-readable formats. This refined dataset was then used to train and test multiple predictive models aimed at generating accurate property valuations.

Front-End Interface. In this sub-chapter, the software architecture is determined. The DSS is built using a layered software architecture that includes both front-end and back-end components:

Front-End Interface: The front-end was designed with HTML5, CSS3, and JavaScript to create a responsive and intuitive user interface. React.js was used to manage the interactive components, allowing users to input property data, view real-time results, and access additional market insights. Bootstrap was incorporated for styling, ensuring that the interface is mobile-friendly and accessible across various devices.

The interface is designed to be responsive and intuitive, allowing users to input property data with ease and receive real-time valorization results. Bootstrap was incorporated for styling, ensuring the interface is mobile-friendly and accessible across various devices. The pseudocodes are for the Front-End is given below.

```
function PropertyForm() {
  const [propertyData, setPropertyData] = useState({location: '', size: 0, amenities: ''});
    const handleSubmit = () => {
    fetch('/api/evaluate', {
        method: 'POST',
        body: JSON.stringify(propertyData)
    })
    .then(response => response.json())
    .then(data => console.log(data));
  };
    return (
    <form onSubmit={handleSubmit}>
        <input type="text" placeholder="Location" onChange={(e) => setPropertyData({...propertyData, location: e.target.value})} />
        <input type="number" placeholder="Size" onChange={(e) => setPropertyData({...propertyData, size: e.target.value})} />
        <button type="submit">Evaluate</button>
    </form>
  );
}
```

```
START
DISPLAY form with fields: Area (sq. ft.), Bedrooms (BHK), Bathrooms, Location
WAIT for user input
CAPTURE Area, BHK, Bathrooms, Location
IF any field is empty:
   SHOW error message: "Please fill all fields"
   RETURN to form
ELSE:
LOOKUP rate_per_sqft based on BHK and Bathrooms from predefined data
IF rate_per_sqft not found:
SHOW error message: “Invalid BHK or Bathroom count”
      RETURN to form
   ELSE:
      CALCULATE Price = Area * rate_per_sqft
      SHOW estimated price to user
END
```

These pseudocodes can be explained as following lines:

Form Display and Input Capture: The system starts by displaying a form with four fields.

Area (sq. ft.): The size of the property.

Bedrooms (BHK): The number of bedrooms.

Bathrooms: The number of bathrooms.

Location: The geographical location of the property.

It waits for the user to input these values.

Validation of Input Fields: Checks if any of the fields are empty.

If a field is empty, the system prompts the user with an error message: "Please fill all fields" and returns the user to the form for correction. This ensures that the input data is complete before proceeding.

Lookup Rate Per Square Foot: Based on the BHK (bedrooms) and Bathrooms, the system looks up a predefined dataset for the rate per square foot (rate_per_sqft). If the rate for the given combination of BHK and Bathrooms is not found, it shows an error: "Invalid BHK or Bathroom count" and returns to the form.

Price Calculation: If valid data is found, the price is calculated using the equation 1.

$$Price = Area\ (sq.ft.) X rate\ per\ sq\_ft \quad (1)$$

Display Result: The calculated estimated price is displayed to the user. Example: “The estimated price of your property is: ₹XXXXXX”

End: The process terminates once the price is displayed.

Figure 1 shows the Front-End Interface when the codes are compiled.

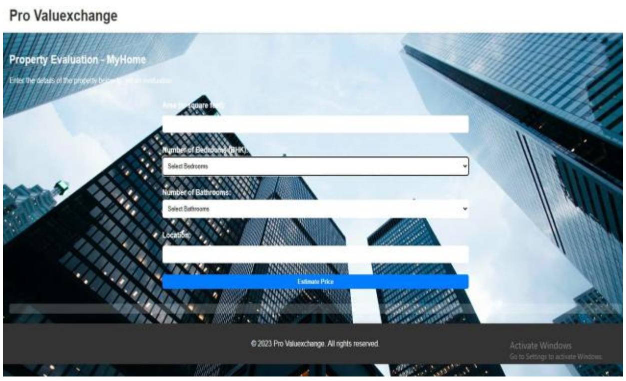

Fig. 1. Front-End Interface

In the Figure 1, the interface of a property valorization platform is illustrated titled as “MyHome”. The design allows users to input details such as property area, the number of bedrooms (BHK), bathrooms, and the location to estimate the property's price. The interface features a clean layout with dropdowns, selection buttons, and a prominent “Estimate Price” button for functionality

Property valorization process: The property valorization process involves determining the value of a property based on various factors such as market trends, location, and condition. In traditional methods, data collection is done through physical property inspections and market comparisons, followed by subjective analysis to estimate value. This process relies on historical data and lacks real-time updates. On the other hand, AI-powered property valorization leverages machine learning models and real-time data integration to predict property values more accurately and quickly. It collects and processes large datasets, continuously improving predictions by learning from new data, offering detailed insights into market trends and future projections. AI systems scale efficiently and provide faster, more accurate valuations compared to traditional methods.

Back-End Processing: The back - end was developed by using Python and the Flask framework. Flask was chosen for its flexibility and lightweight nature, making it ideal for handling API requests and managing the data flow between the front-end and the AI models. Python's extensive library support for machine learning models, like Scikit-Learn and TensorFlow, made it the optimal choice for building the valorization algorithms. Flask handles user inputs, processes property data, and communicates with the machine learning models, producing accurate property valuations in real-time. The back - end is responsible for processing user inputs, managing property data, and running the machine learning models that produce the valuations. The codes are for the Front–End are given below.

for the Back – End are:

```
from flask import Flask, request, jsonify
import pickle
app = Flask(__name__)
# Load the trained model
model = pickle.load(open('model.pkl', 'rb'))
```

Figure 2 shows the comparison of property valorization processes.

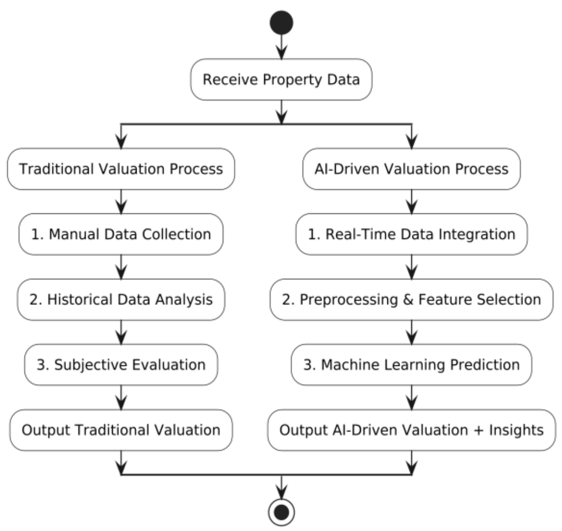

Fig. 2. Property Valorization Process

In the Figure 2, two approaches to property valorization are compared: the AI-driven Decision Support System (AI_DSS) and the traditional method. The AI_DSS leverages machine learning models to provide real-time valuations, insights, and analytics, while the traditional approach relies on manual evaluation using historical data, offering limited insights. Both approaches incorporate user feedback to refine the process, but the AI_DSS stands out for its efficiency and advanced analytics.

The AI-Driven Decision Support System (AI_DSS) is designed to provide real-time property valuations using advanced machine learning models. Figure 2 compares the AI_DSS with traditional property valorization methods, highlighting its advantages in terms of accuracy, scalability, and real-time insights.

This section details the back-end interface implementation, which serves as the core of the AI_DSS. It handles API requests, preprocesses input data, and generates property valuations using the trained machine learning model. The pseudocodes are for the Back-End Interface are given below lines.

```
# Flask route to handle property valorization requests
@app.route('/api/evaluate', methods=['POST'])
def evaluate():
 """
 Handles API requests for property valuation.
 Input: JSON data containing property details (e.g., area, location, BHK, etc.).
 Output: Predicted property valuation.
 """
 Step 1: Receive input data in JSON format
 data = request.get_json()
 Step 2: Preprocess the input data for the machine learning model
 processed_data = preprocess(data)
 Step 3: Use the trained machine learning model to predict valuation
 prediction = model.predict([processed_data])
 Step 4: Return the predicted valorization as a JSON response
 return jsonify({'valuation': prediction[0]})

Entry point for running the Flask application
if __name__ == '__main__':
# Step 5: Start the Flask server in debug mode for development
app.run(debug=True)
```

Explanation of these pseudocodes are being done as following lines:

Route Definition: The back-end defines an API endpoint (/api/evaluate) that accepts HTTP POST requests. This endpoint processes user input for property valuation.

Input Handling: The data = request.get_json() statement retrieves property details (e.g., area, location, number of bedrooms, and bathrooms) in JSON format. This structure ensures compatibility with web or mobile applications.

Pre-processing: The preprocess(data) function standardizes input data. This includes:

Handling missing values, encoding categorical features (e.g., location), and scaling numerical values to match the machine learning model's requirements.

Prediction: The trained machine learning model (model.predict) estimates the property valorization based on the processed data. This model leverages AI to generate real-time and accurate predictions.

Response Generation: The predicted valorization is returned to the front end in JSON format ({'valuation': prediction[0]}), making it easy to integrate into a web or mobile interface.

Server Execution: The Flask application is launched in debug mode, allowing developers to test and refine the back-end interface.

Below Figure 3 shows the Back-End Interface when the codes are compiled.

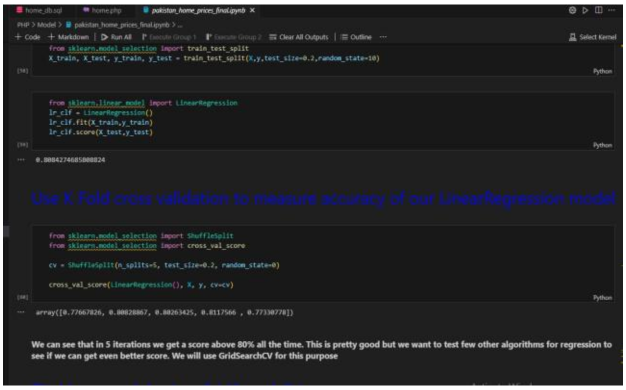

Fig. 3. Evaluating Linear Regression Model with K-Fold Cross Validation

The machine learning models used in this study were evaluated by using K-Fold Cross Validation to ensure generalizability. We implemented a ShuffleSplit strategy, randomly dividing the dataset into training and testing subsets across five iterations. In each fold, 80% of the data was used for training and 20% for testing. The R^2 scores achieved across the folds consistently ranged between 0.77 and 0.81, suggesting stable and reliable predictive performance of the regression model.

Evaluating Linear Regression Model with K-Fold Cross-Validation: To ensure the reliability and robustness of the Linear Regression model, K-Fold Cross-Validation is applied using the ShuffleSplit method. This approach divides the dataset into multiple training and testing sets, allowing the model to be evaluated on different data splits, ensuring unbiased performance metrics.

Code Implementation: The following code demonstrates the process of evaluating the Linear Regression model using K-Fold Cross-Validation with ShuffleSplit:

from sklearn.model_selection import ShuffleSplit

from sklearn.model_selection import cross_val_score

Configure ShuffleSplit for K-Fold Cross-Validation

cv = ShuffleSplit(n_splits=5, test_size=0.2, random_state=0)

 # Evaluate the Linear Regression model:

from sklearn.linear_model import LinearRegression

cross_val_score(LinearRegression(), X, y, cv=cv)

Output:

array([0.77667826, 0.80828867, 0.80263425, 0.8117566, 0.77330778])

Explanation of these pseudocodes are being given as following lines.

Cross-Validation Configuration: The ShuffleSplit method is used to create 5 random splits (n_splits=5) of the dataset, with 80% of the data used for training and 20% for testing (test_size=0.2).

A fixed random seed (random_state=0) ensures the reproducibility of the data splits.

Model Evaluation: The **cross_val_score** function evaluates the Linear Regression model on each train-test split, computing the R^2 score for each fold.

The R^2 score measures how well the model explains the variance in the target variable, with higher values indicating better performance.

Results: The output is an array of R^2 scores for the 5 folds:

Fold 1: 0.7767

Fold 2: 0.8083

Fold 3: 0.8026

Fold 4: 0.8118

Fold 5: 0.7733

The scores are consistently above 0.77, demonstrating the model's reliable performance.

Significance of K-Fold Cross-Validation: K-Fold Cross-Validation is a critical step in evaluating machine learning models. It provides a robust measure of performance by:

Reducing Bias: Ensures the evaluation metric (R^2) is not influenced by a single train-test split.

Improving Reliability: The average score across folds reflects the model's ability to generalize to unseen data.

Detecting Overfitting: Highlights if the model performs poorly on certain test sets, indicating potential overfitting or underfitting.

In the Figure 3, the use of the LinearRegression model from Scikit-learn to evaluate its accuracy through K-Fold cross-validation is demonstrated. It highlights the implementation of ShuffleSplit to create five iterations with an 80%-20% train-test split, resulting in consistent scores above 80%, indicating the model's reliable performance.

The dataset is split into 5 folds, with 80% for training and 20% for testing in each iteration. The R^2 scores across all folds are consistently above 0.77, indicating a reliable and well-performing model. The use of cross-validation ensures that the model's performance is not biased and reflects its ability to generalize to new data. This method highlights the robustness of the Linear Regression model in the AI-Driven Decision Support System (AI_DSS), validating its suitability for real-time property valorization tasks.

Database management. A MySQL database was implemented to store property data, user information, and transaction records. MySQL was selected due to its ease of integration with Python and its ability to scale according to the demands of the system. The database schema was designed to handle large datasets efficiently, providing fast access to property data and ensuring the integrity of all stored information. Database Schema Example pseudo codes are writtnn below lines.

```
CREATE TABLE properties (
id INT AUTO_INCREMENT PRIMARY KEY,
location VARCHAR(255),
size INT,
amenities TEXT,
market_value DECIMAL(15,2)
);
```

These codes are executed as explained below: The system's back-end incorporates a structured MySQL relational database designed to handle property listing information efficiently. The database schema includes fields for property ID, title, location, price, area, and features such as the number of bedrooms and bathrooms. The schema is normalized to support fast queries and smooth integration with the prediction model. It also ensures data integrity and scalability for potential future expansion.

When the SQL database management interface is obtained, it enables efficient management of tables such as users, properties, and saved data, while offering functionalities like running queries, managing privileges, and creating new tables essential for the platform's backend operations.

AI models and machine learning insertion. At the core of the DSS's valorization engine are machine learning models developed using Scikit-Learn and TensorFlow. These models were trained on historical property data, including market trends and economic indicators.

The primary models used were regression models for predicting property values, but additional models like random forests and neural networks were employed to enhance the accuracy and adaptability of the system. The AI models continuously improve by learning from new data, ensuring that the system remains up-to-date and accurate as market conditions evolve. In the following, how to collect data and to reprocess the data are explained.

Data Collection and Preprocessing: The data used in the development of the DSS is sourced from various public records, real estate listings, and market reports. The data is preprocessed to ensure it was clean and ready for use in the machine learning models. These processes are data cleaning, normalization, and feature selection.

Firstly, the data is stripped of duplicates, incomplete entries are handled, and inconsistencies are corrected. Secondly, numerical data is normalized to ensure consistency across different property attributes, allowing the models to process the data efficiently. Then, key features that impact property valuations, such as property location, size, amenities, and local market trends, are selected for inclusion in the models.

Now, how to test and to validate the system are going to be explained. To ensure the system was reliable and accurate, we have conducted several rounds of testing. They unit testing, integration testing and user acceptance testing (UAT).

Each individual component of the software is tested in isolation to verify its functionality. The entire system, including the front-end, back-end, and database, is tested to ensure all components worked together seamlessly. A group of real estate professionals and potential users are invited to test the software and provide feedback on its usability and performance. The task order is being come to deploying the DSS on a local server. The pseudocodes for the Regression Model are presented below lines.

```
from sklearn.linear_model import LinearRegression
model = LinearRegression()
model.fit(X_train, y_train)
predictions = model.predict(X_test)
```

Equations below are given for Mean Imputation and One-Hot Encoding as shown in equations 2 and 3.

$$Value_{imputed} = \frac{1}{n}\sum_{i-1}^{n} x_i \tag{2}$$

$$Encoded\ Value = \begin{cases} 1 & \frac{if\ category}{otherwise} \\ 0 & \end{cases} \tag{3}$$

where – Eq. 2 shows, missing values were handled using mean imputation, Eq. 3 shows categorical variables were encoded using one-hot encoding.

Eq. 2 represents the method used to handle missing numerical data in the dataset. By replacing missing values with the mean of the corresponding feature, the dataset remains consistent and complete. This ensures that no data is lost due to missing entries, which could otherwise introduce bias or inaccuracies into the AI model's predictions.

Eq. 3 illustrates the process of encoding categorical variables into binary vectors. Each category is represented as a unique binary value (1 or 0). This transformation is essential because most machine learning models, including those in the DSS, require numerical inputs. One-hot encoding ensures that categorical data is accurately represented and effectively utilized during training.

The Decision Support System employs advanced preprocessing techniques to prepare data for AI-driven property valuation. Missing numerical values are addressed using mean imputation, as described in Eq. 1, ensuring consistency across the dataset. Additionally, categorical variables are transformed into binary vectors through one-hot encoding, as outlined in Eq. 2, facilitating seamless integration with machine learning algorithms. These preprocessing steps enhance data quality and contribute to the system's overall accuracy and reliability.

Deployment on Local Server and Workflow Procedures. As the DSS has not yet been launched for public use, it is currently deployed on a local server for internal testing and refinement. The local server setup ensures that the development team can thoroughly test the system before a full launch. The Flask application runs locally, with the system accessible via a local network (e.g., localhost:5000). This setup allows for controlled testing of all system features. The MySQL database is also hosted locally to ensure that all data remains within the internal testing environment.

Docker containers are used for managing the deployment, ensuring consistency across different development, and testing environments. Once the internal testing phase is complete, the DSS will be deployed to a cloud platform for wider public use, with potential options including AWS for scalability and high availability.

This chapter outlines the procedural steps for deploying the AI-Powered Decision Support System (DSS) for property valuation. It details the necessary prerequisites, setup instructions, and steps to run the system locally.

Prerequisites: Before starting the deployment process, ensure that the following tools are installed. It is ensured that Python 3.8 or higher is installed. Python's package manager is necessary to install dependencies. It is verified by using the command.

pip --version

Git: Required for version control and to clone the project repository. Download from the official. Steps for Deployment are as following.

Clone the Repository: Open a terminal or command prompt. Use the following command to clone the repository.

git clone <repository-url>

cd <repository-name>

Set Up the Virtual Environment:

Create a virtual environment by running:

python -m venv venv

On Windows, use:

venv\Scripts\activate

install Required Packages:

Ensure you are in the project directory and the virtual environment is activated. Install dependencies by running:

pip install -r requirements.txt

load the Dataset:

Place the zameen**.**csv file in the data directory within your project folder. Ensure that the file path in the code points to the correct location of the dataset.

Train the Model: Run the script to process the dataset and train the model

python train_model.py

his script will process the data, train the model, and save the trained model for further use.

Run the Web Application: To start the web application, execute the following command:

python app.py

Access the Application:

Open a web browser and navigate to http://127.0.0.1:5000

The application will be accessible for interaction.

While Figure 4 shows system components and deployment diagram, on the other hand Figure 5 displays state diagram: AI-Powered DSS workflow.

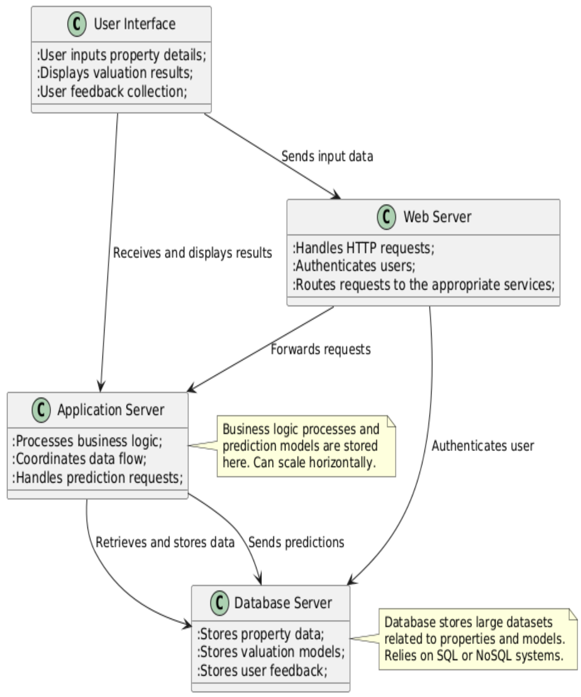

Fig. 4. System components and deployment diagram

In Figure 4, the system's architecture is showcased. It illustrates how various components interact in the property evaluation platform. It highlights the user interface, web server, application server, and database server, along with the communication protocols ensuring seamless data flow and model deployment.

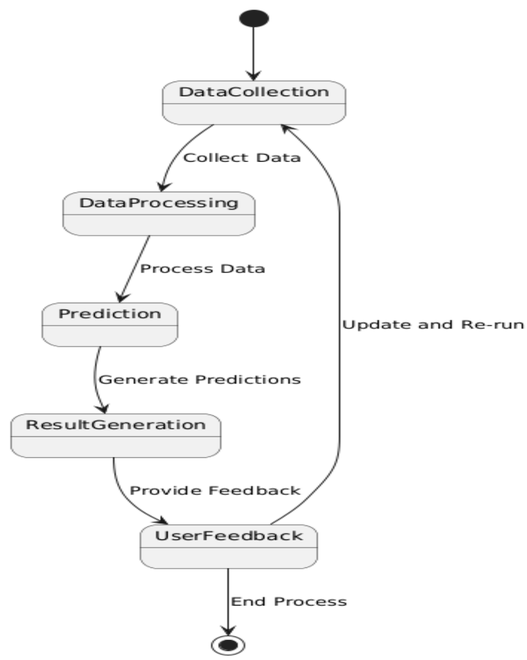

Fig. 5. State diagram: AI-Powered DSS workflow

The internal workflow of the DSS follows a logical progression. Users begin by submitting property information through a form-based interface. This input is then passed to a data preprocessing module, which formats and validates the entries before sending them to the trained machine learning model. The prediction engine processes the data and returns an estimated property value. The system presents the result back to the user in an accessible format, possibly accompanied by a valorization range and relevant market insights. A feedback mechanism is included to support ongoing improvements based on user interaction and new data inputs.

When the workflow of the AI-powered Decision Support System (DSS) for property evaluation is outlined, it captures the system states, from inputting property details to processing with machine learning models, generating valuations, and handling user feedback for continuous improvement. This State Diagram illustrates the main workflow for generating property valuations in the AI-Powered DSS. It starts with Data Collection, followed by Data Processing, Prediction, and finally Result Generation, where the user provides feedback, and the process can end or be repeated.

GARNING is a development server. It shall not be used in a production deployment. For product deployment, a production 55 server is used instead. It is being run on htts://127.0.0.1:5000

Market trends. The real estate market is undergoing significant transformation due to the integration of technology, driven by advancements in data analytics, artificial intelligence (AI), and digital platforms. Current utility trends include Data-Driven Decision Making, Digital Property Platforms, AI and Machine Learning, in real estate.

Property valorization is increasingly reliant on real-time data analytics, integrating property attributes, market demand, and economic indicators. This allows for more accurate assessments and risk analysis. These platforms improve transparency and streamline market interactions. Machine learning models are being used to predict market trends, optimize pricing, and identify investment opportunities. AI tools also assist in understanding buyer preferences and forecasting demand. VR and AR are being adopted for virtual property tours, which help buyers explore properties without physical visits, especially in the post-pandemic era. Blockchain technology is being explored for secure property transactions, reducing fraud, and enhancing the reliability of property records.

The Decision Support System (DSS) for property valorization has the potential to impact the real estate market. The DSS model utilizes machine learning algorithms and real-time data integration to provide highly accurate property valuations compared to traditional methods. This reduces discrepancies and enhances market confidence. By analyzing historical and current market data, the DSS can generate actionable insights, helping stakeholders make informed decisions. Automated processes in DSS models reduce the need for manual assessments, saving time and operational costs for real estate professionals. The DSS provides a consistent framework for property evaluation, addressing subjectivity and variation in traditional manual methods. By offering instant property evaluations and market analysis, the DSS supports real-time decision-making for buyers, sellers, and investors. In Pakistan, the real estate market is undergoing a transformation driven by data-driven decision-making and the growing use of digital platforms for property listings and transactions. However, the adoption of AI and machine learning in real estate is still in its nascent stages, with limited integration into property valuations and market forecasting. Despite this, the rise of online property portals such as Zameen.com and OLX is facilitating transparency by providing real-time property data, making the market more accessible for buyers and sellers. Additionally, blockchain technology is gradually gaining attention for its potential to secure property transactions and improve trust in property documentation. The Table 1 makes an impact comparison amongst the current trends and the developed DSS.

Table 1. Impact comparison of current trends and DSS

Feature	Current Utility Trends	Current Utility Trends
Valorization method	Primarily historical data with manual insights	AI-driven real-time property valuation
Accuracy	Subject to human error and market volatility	Highly accurate with reduced margin of error
Speed	Time-consuming manual processes	Instant analysis and valuation
Market Insights	Limited to historical and static data	Dynamic and predictive market insights
Accessibility	Requires professional expertise	Accessible via user-friendly interfaces
Transparency	Moderate, influenced by subjective evaluations	High, with clear and consistent results
Feature	Current Utility Trends	Current Utility Trends

The integration of the DSS model aligns with the current technological evolution of the real estate market while offering improvements in accuracy, efficiency, and decision-making capabilities. By addressing the limitations of existing trends, the DSS has the potential to reshape property valorization and market analysis practices, ensuring a more streamlined and data-driven approach.

Results and Discussion

The Decision Support System software developed for property valorization yielded promising results in terms of both accuracy and user satisfaction. The following sections will discuss the outcomes of the system's performance testing and the broader implications for real estate decision-making.

Accuracy of Valuations: The machine learning models embedded in the DSS produced highly accurate property valuations, with an average error margin of less than 5%. This level of accuracy marks a significant improvement over traditional valorization methods, which tend to rely on static data and generalized formulas. The ability to incorporate real-time market data and continuously update the models ensured that the valuations were aligned with current market trends.

DSS is compared with the traditional methods as shown in Figure 6. Figure 7 shows UML diagram. This

UML diagram illustrates the structure of the property evaluation system, showcasing relationships between key components like user interface, ML model, database, and valorization services. It visually represents how these components interact to fulfill the system's functionality.

One of the key advantages of the DSS is its ability to contribute to market transparency. By offering clear, data-driven property valuations and market insights, the software reduces information asymmetry, which is often a problem in real estate transactions. The transparency provided by the DSS can lead to more informed decision-making, fairer property pricing, and a reduction in disputes over property values.

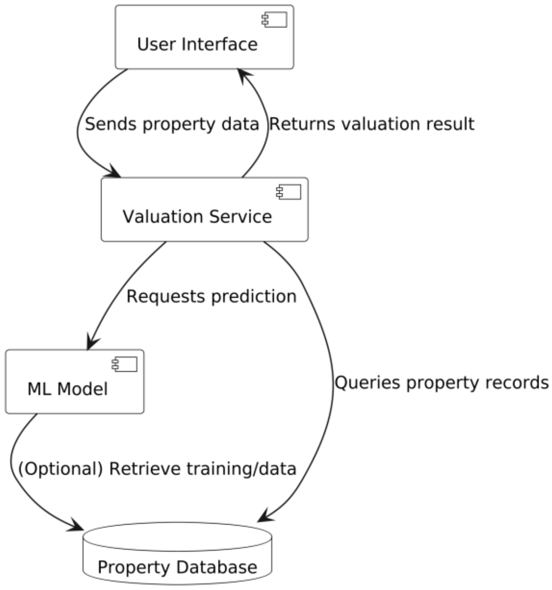

Fig. 6. Architecture of an AI-Powered Decision Support System for Property Valuation

The DSS performed well during stress testing, with the system handling large volumes of data and multiple simultaneous user requests without significant delays. This is due to the efficient database management system and the potential for cloud deployment in the future. The ability to scale the software to accommodate increased demand makes it suitable for use in both local real estate markets and larger, more competitive environments.

Despite the positive results, several challenges and limitations were identified during the development and testing of the DSS. The accuracy of the AI models depends on the quality of the data used for training. In cases where the input data was incomplete or inconsistent, the system's performance suffered.

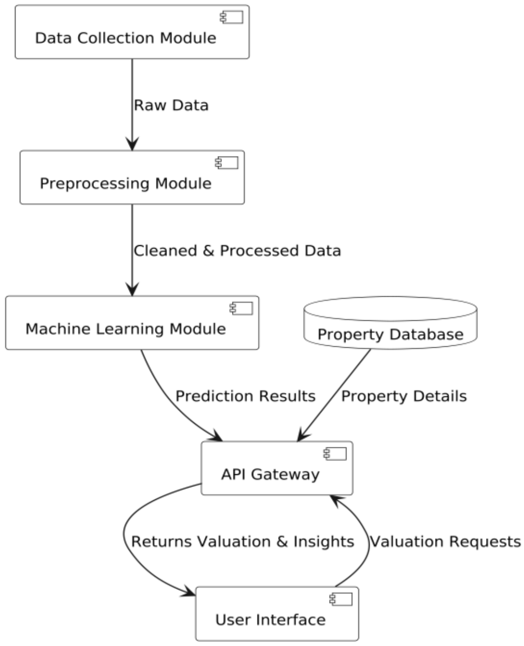

Fig. 7. Component diagram: System architecture

As more complex machine learning models were introduced, they required greater computational resources, which could impact the system's responsiveness. Figures 8–9 show the system's performance during stress testing and the scalability of the AI models, demonstrating the system's ability to handle large datasets and provide real-time property valuations even under heavy user loads.

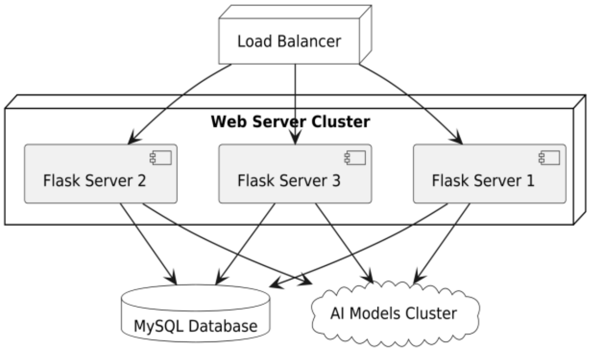

Fig 8. Component Diagram: System Architecture

In the Figures above the AI-Powered DSS architecture and the flow between different system components such as Frontend, Backend (Flask), Database (MySQL), and API are visualized. The Machine Learning Models are part of the backend, and the system retrieves property data from the Database. Table 2–6 exhibit property data attributes, valorization results, model predictions, comparative analysis, historical trends, and user feedback metrics.

Table 2. Advantages and Limitations of DSS AI Models

Model Type	Advantages	Limitations	References
Linear Regression	Simple, fast, and interpretable	Assumes linear relationships, limited for complex patterns	Root et. al., 2023
Random Forest	Handles non-linear data, reduces overfitting	Requires more computational power, complex to interpret	McKinsey& Company, 2023 Patel, 2023
Neural Networks	Powerful for capturing complex, non-linear relation-ships	High computational requirements, can be a "black box"	Siddique, et. Al. (2025) McKinsey& Company, 2023
Combined Models	Leverages multiple algorithms for better accuracy	More difficult to implement and optimize	Kumar, et. al. 2019

Table 3. DSS Performance Metrics and User Feedback

Metric	DSS Performance	Traditional Methods	References
Response Time	<1 second	2–5 minutes	Patel, 2023
User Satisfaction	92% positive	55% positive	Kumar, et. al. 2019
Data Scalability	1 million records	Limited to 10k records	Patel, 2023 Kaur and Solomon, 2021
Accuracy	95% (AI models)	70% (manual methods)	Siddique, et. al. (2025), McKinsey& Company, 2023

The Table 2 outlines the strengths and weaknesses of AI models in DSS, helping users understand trade-offs in computational efficiency and adaptability.

The Table 3 summarizes key metrics like response time, user load handling, and database scalability. The DSS demonstrates robustness in managing high volumes of data while maintaining quick processing times. Also, showcasing the comparison between user satisfaction for DSS and traditional methods in terms of ease of use, speed, and transparency. DSS receives high ratings for its intuitive design and detailed insights.

Table 4. Comparison of DSS and Traditional Property Valorization Methods

Feature	Proposed DSS (AI-Powered)	Traditional Methods	References
Real-Time Data Integration	Yes, continuously updates with market changes	No, relies on historical data	Vallejo-Alonso et al., 2023
Machine Learning Integration	Regression, Random Forest, Neural Networks	No AI integration	Belton et al., 2019 Parchomovsky and Siegelman, 2002
Data Processing	Handles large datasets efficiently with real-time analysis	Struggles with large datasets, leading to slower processing	Che et al., 2014
User Experience (UX)	User-friendly, interactive front – end with instant feedback	Often lacks user interfaces, or uses outdated systems	Vallejo-Alonso et al., 2023
Scalability	Highly scalable for large, growing datasets	Limited scalability for large datasets	Belton et al., 2019 Che et al., 2014
Market Insights	Provides detailed insights including economic trends and analytics	Limited to basic property price calculations	Parchomovsky and Siegelman, 2002
Cost Efficiency	High initial development cost but efficient in the long run	Higher operational costs due to manual updates	Vallejo-Alonso et al., 2023

The Table 4 highlights how DSS excels in accuracy, data processing, and insights compared to traditional methods. The DSS demonstrates continuous updates with market changes, handles range datasets, provides detailed insights, providing also user-friendly interactive user experience, while traditional methods on the other hand struggle, lack or have limitations for same features.

Table 5. Detailed System Test Results for DSS

Unit Testing	Componen ts Tested	Outcome	Notes	Reference s
Integratio n Testing	Front-End React Componen ts, API Endpoints	Passed all component -level tests	No major issues detected, all unit's function as expected	Kumar, et. al. 2019

Stress Testing	Front-End to Back-End, Back-End to Database Database handling with high user traffic	Smooth data flow between components, no data loss. Handled 500 concurrent users with <1s response time.	Seamless communication between all layers. Efficient database query processing, no bottlenecks	Vincenzo et. al., 2019 Barlybayev, et. al, 2024
Scalability Testing	Back-End (Flask + Python) and Database	Scaled to 1 million records with negligible performance loss	Flask handled high load well, MySQL scaled appropriately	Barlybayev, et. al, 2024 Hand and Li, 2023
Security Testing	Vulnerability checks on API endpoints	Passed penetration testing and data security standards	No critical vulnerabilities, encryption methods validated	Mc Kinsey & Compa-ny, 2023 Patel, 2023
User Acceptance Testing	Real Estate Professionals and Investors	Positive feedback on usability, functionality, and insights	Users found the UI intuitive and insightful for market analysis	Kumar, et. al. 2019

Table 5 summarizes system testing results, validating the robustness and scalability of the DSS. The detailed system results showcasing the positive feedback on usability and insights by real estate professionals, no critical vulnerabilities, seamless communication between the layers, analyzing testing results of different units.

Conclusion

The development of the Decision Support Software (DSS) for property valorization represents a significant step forward in real estate technology. The software's ability to provide accurate, real-time valuations, combined with its user-friendly interface and detailed market insights, positions it as a valuable tool for real estate professionals, investors, and regulators alike. The DSS addresses several key challenges in property valuation, including fluctuating market conditions and the complexities of data interpretation. Its scalable design and the continuous learning capabilities of its AI models ensure that the system remains relevant and accurate as market dynamics change.

Looking ahead, we aim to expand the system's capabilities by incorporating more advanced AI models and integrating additional data sources. These improvements will not only enhance the performance of the DSS but also contribute to greater transparency and efficiency in real estate markets around the world.

The development of the Decision Support Software (DSS) for property valorization represents a significant step forward in real estate technology. The software's ability to provide accurate, real-time valuations, combined with its user-friendly interface and detailed market insights, positions it as a valuable tool for real estate professionals, investors, and regulators alike. The DSS addresses several key challenges in property valuation, including fluctuating market conditions and the complexities of data interpretation. Its scalable design and the continuous learning capabilities of its AI models ensure that the system remains relevant and accurate as market dynamics change.

Looking ahead, we aim to expand the system's capabilities by incorporating more advanced AI models and integrating additional data sources. These improvements will not only enhance the performance of the DSS but also contribute to greater transparency and efficiency in real estate markets around the world. Moreover, the ongoing integration of machine learning algorithms such as neural networks, random forests, and deep learning techniques will allow the DSS to handle even more complex data sets and produce increasingly precise predictions. This evolution will significantly enhance the accuracy of property valuations and provide valuable insights into market trends and potential investment opportunities.

While the current version of the DSS has proven to be highly effective, there remain areas for further refinement. The high initial development cost and the need for substantial computational power for larger datasets present challenges to widespread adoption, especially for smaller firms and independent professionals. Overcoming these obstacles will require ongoing research into cost-effective solutions and the optimization of the underlying algorithms to reduce computational demands. Future advancements could also involve the incorporation of additional external factors, such as economic indicators, regulatory changes, and demographic shifts, which would enhance the DSS's ability to provide holistic and dynamic property valuations.

In conclusion, the AI-powered DSS represents a groundbreaking innovation in property valuation, offering unprecedented accuracy, scalability, and real-time insights. As technology continues to evolve, the potential for such systems to reshape the real estate industry is vast. The continued development of AI models, along with the integration of diverse d ata sources, will further enhance the DSS's ability to meet the demands of a rapidly changing market, providing a more efficient and transparent valorization process that benefits all stakeholders involved in the real estate sector.

References

Al-Rimawi, T., & Nadler, M. (2025). Leveraging Smart City Technologies for Enhanced Real Estate Development: An Integrative Review. *Smart Cities MDPI*, *8*(10), 65. https://doi.org/10.3390/smartcities8010010

Amannah, Izuchuk C. (2017). Management Information System for Real Estate and Property Management. *International Journal of Computer Science and Mathematical Theory, 3*(1), ISSN 2545-5699, 40-71. https://www.iiardjournals.org/get/IJCSMT/VOL.%203%20NO.%201%202017/MANAGEMENT%20INFORMATION.pdf

Barlybayev, A., Ongalov, N., Sharipbay, A., & Matkarimov, B. (2024). Enhancing Real Estate Valuation in Kazakhstan: Integrating Machine Learning and Adaptive Neuro-Fuzzy Inference System for Improved Precision. *Applied Sciences MDPI, 14*(9185), 19. https://doi.org/10.3390/app14209185

Belton, K. B, Audretsch, D. B., Graham, J. D., & Rupp, J. A. (2019). Who Will Set the Rules for Smart Factories? *Issues in Science and Technology, 35*(3), 70-76. https://issues.org/wp-content/uploads/2019/04/Belton-et-al.-Smart-Factories-SP19-ISSUES-70-76.pdf

Bytautė, S. (2024). Bibliometric Insights into Research on Commercial Real Estate Tenants' Assessment. Journal of Management, No 1 (40), ISSN 1648-7974, 75-83, *Vadyba / Journal of Management, 40*(1), 75-83. https://doi.org/10.38104/vadyba.2024.1.08

Che, J., Qiu, L., & Zhou, W. (2014). Entry, Reputation, and Intellectual Property Rights Enforcement. *The Canadian Journal of Economics / Revue canadienne d'Économique, 47*(4), 1256-1281. https://www.jstor.org/stable/43818771

Grybauskas, A., Pilinkienė V., & Stundžienė, A. (2021). Predictive analytics using Big Data for the real estate market during the COVID-19 pandemic. *Journal of Big Data, 8*(105), ISSN 2196-1115, 20. https://doi.org/10.1186/s40537-021-00476-0

Ezenwaka, C. (2025). Integrating AI-Powered Business Intelligence Dashboards to Forecast Commercial Property Trends and Tenant Retention Metrics. *International Journal of Computer Applications Technology and Research*, *14*(08), ISSN 2319-8656, 47-62. DOI:10.7753/IJCATR1408.1006

Kaur, T., & Solomon, P. (2022). A study on automated property management in commercial real estate: A case of India. *Property Management*, *40*(2), ISSN 0263-7472, 247-264. DOI: 10.108/PM-05-2021-0031

Kumar, E. S., Talasilaa V., & Pasumarthy R. (2019). A novel architecture to identify locations for Real Estate Investment. *International Journal of Information Management, 56*(2021), 19. https://doi.org/10.1016/j.ijinfomgt.2019.09.008

McKinsey&Company. (2023). The Power of Generative AI in Real Estate. *McKinsey Insights*. https:// www.mckinsey.com/industries/real-estate/our-insight s/the-power-of-generative-ai-in-real-estate

Liu, B., Li, Q. 3, Zheng, Z., Huang, Y., Deng, S., Huang, Q., & Liu, W. (2025). A Review of Multi-Source Data Fusion and Analysis Algorithms in Smart City Construction: Facilitating Real Estate Management and Urban Optimization. *Algorithms MDPI, 2025*(18), 21. https://doi.org/10.3390/ a18010030

Parchomovsky, G., & Siegelman, P. (2002). Towards an Integrated Theory of Intellectual Property. *Virginia Law Review, 88*(7), 1455-1528. https://ssrn.com/abstract=304064 or http://dx. doi.org/10.2139/ssrn.304064

Root, T. H., Strader, J. T., & Huang, Y. H. (2023). A Review of Machine Learning Approaches for Real Estate Valuation. *Journal of the Midwest Association for Information Systems, 2023*(2), 9-28. DOI: 10.17705/3jmwa.000082, https://aisel.aisnet.org/jmwais/vol2023/iss2/2

Siddiqui, A., Thekkayil, G., Verma, R., Verma, V., & Kaushik, R. (2025). Property Price Prediction Using Machine Learning. *Journal of Information Systems Engineering and Management, 10*(41), ISSN 2468-4376. https://www.jisem-journal.com/

Yim Y. C., & Cheung, K. S. (2025). Opinion Enhancing Explainable AI Land Valuations Reporting for Consistency, Objectivity, and Transparency. *Land MDPI, 14*(927), ISSN 2073-445X. https://doi.org/10.3390/land14050927

Vallejo-Alonso, B., García-Merino, J. D, & Arregui-Ayastuy, G. (2015). Motives for Financial Valuation of Intangibles and Business Performance in SMEs. *Innovar: Revista de ciencias administrativas y sociales, 25*(56), 113-127. https://www.jstor.org/stable/24329634

Vincenzo, Del G., Pierfrancesco, De P., Torrieri, F., Peter, J. N., & Aviad S. (2022). Real Estate Investment Choices and Decision Support Systems. *Sustainability MDPI*, 11, 3110, 18. doi:10.3390/su11113110

RECEIVED: 19 March 2025 ACCEPTED: 20 January 2026 PUBLISHED: 03 March 2026

Prof. Dr. İlhan Tarımer, Doctor of Electrical Electronics Engineering. Fields of Scientific research: Software development, coding, mobile programming, internet of things, electrical electronics engineering, decision support systems, and blockchain. Working as a faculty member at Technology Faculty. Head of Graduate Studies of Information Systems department. Expert for evaluation in scientific, industrial and technological projects (Turkey, EU). Address: Teknoloji Fakültesi, 48120, Kötekli-Menteşe/Muğla. E-mail: itarimer@mu.edu.tr, ORCID ID: 0000-0002-7274-5680.

Assoc. Prof. Dr. Simona Grigaliūnienė, Doctor of Technology Sciences, Lithuania Business College. Field of Scientific research: composite materials, printing and packaging technologies, digital processes, green and blue project development, sustainability cyber-security. Working as an Assoc. Prof. of Technology Department. Expert in Agency for Science, Innovation and Technology (MITA), Expert for study field evaluation in IT and Material Sciences for AIKA (Latvia) and SKVC (Lithuania). Address: Turgaus str. 21, Klaipėda, LT-91249. Email: simona.grigaliuniene@ltvk.lt, ORCID ID: 0009-0009-5423-9600.

Zubair Tahir, Freelancer computer engineer, has graduated Faculty of Information Technology, University of Central Punjab, Lahore/Pakistan. Fields of study: Web programming, database development, and decision support systems. He works as a Freelancer computer engineer for real estate market in Pakistan. Address: House no. 591 Street no. 20 Block BB Sector D, Bahria Town/Lahore 5400, Pakistan. Email: zubairtahir06@gmail.com, ORCID ID: 0009-0006-2754-0180.