International Journal of Electrical and Computer Engineering (IJECE)
Vol. 14, No. 6, December 2024, pp. 6619~6627
ISSN: 2088-8708, DOI: 10.11591/ijece.v14i6.pp6619-6627

Neutrosophic enhanced convolutional neural network for occupancy detection: structured model development and evaluation

Ranjeeta Mittal[1], Suresh Kumar[1], Urvashi Chugh[2]
[1]Department of Computer Science and Engineering, School of Engineering and Technology, Manav Rachna International Institute of Research and Studies, Faridabad, India
[2]Department of Information and Technology, Krishna Institute of Engineering and Technology, Ghaziabad, India

Article Info

Article history:

Received Apr 30, 2024
Revised Jul 19, 2024
Accepted Aug 6, 2024

Keywords:

Contradictory information
House occupancy detection
Neutrosophic convolutional neural network
Smart home system
Uncertainty handling

ABSTRACT

This study introduces an advanced convolutional neural network (CNN) model tailored for building occupancy detection that accommodates the inherent uncertainties and contradictory information often encountered in sensor data. By integrating neutrosophic layers into the CNN architecture, we enable the model to effectively handle indeterminacy, vagueness, and inconsistency present in real-world sensor readings. The approach employs neutrosophic convolutional, max-pooling, and logic layers, providing a comprehensive framework for feature extraction and decision-making. Through a structured methodology encompassing data preprocessing, model initialization, training, evaluation, and optimization, we demonstrate the efficacy of the proposed model in accurately detecting occupancy status within residential environments. This enhanced CNN model offers improved accuracy, robustness, and interpretability, thereby facilitating its integration into smart building management systems and building automation applications, to enhance efficiency, comfort, and energy savings.

Corresponding Author:

Ranjeeta Mittal
Department of Computer Science and Engineering, School of Engineering and Technology, Manav Rachna International Institute of Research and Studies
Surajkund Road, Faridabad, Haryana, India
Email: ranjeetamittal@gmail.com

1. INTRODUCTION

Occupancy detection, also known as occupancy sensing or occupancy monitoring, is a fundamental aspect of smart building automation systems, enabling various applications such as intrusion detection, power and devices management, waste management, effectively managing security, parking spaces, emergency exits, and personalized user experiences [1]. The ability to accurately detect occupancy in residential spaces allows for the automation of lighting, heating, ventilation, and air conditioning (HVAC) systems, contributing towards energy savings as well as improved comfort for occupants [2]. In recent years, the proliferation of internet of things (IoT) devices and sensors has facilitated the collection of data from residential environments, offering new opportunities for occupancy detection. These sensors can include passive infrared (PIR) motion sensors, door and window sensors, light sensors, acoustic sensors, and even devices such as laptops, tablets, digital assistants, smartphones, and smart wearable devices [3]. By leveraging the data collected from these sensors, machine learning algorithms can be employed to infer occupancy status and make intelligent decisions in real time.

Convolutional neural networks (CNNs) apparently are a powerful tool in the field of occupancy detection, particularly when dealing with sensor data. CNNs is a specialized technique that belongs to the broader category of artificial neural networks. CNNs are curated to recognize patterns in spatial data. This property makes them an apt choice for applications like image recognition, natural language processing, and time-series analysis [4]. In the context of occupancy detection, CNNs can analyze sensor data streams to identify patterns associated with occupancy and non-occupancy states. This paper presents an enhanced CNN-based approach for occupancy detection using sensor data. The proposed model aims to improve the performance of occupancy detection in smart environments, by improving model accuracy and reliability, thereby enhancing the performance of building automation systems. The remainder of this introduction provides an overview of the significance of occupancy detection, the challenges involved, and the state-of-the-art techniques in the field.

a. Significance of house occupancy detection: Accurate occupancy detection is essential for optimizing energy consumption in residential buildings. By automatically adjusting heating, cooling, and lighting systems based on occupancy status, energy wastage can be minimized, leading to reduced utility bills and environmental impact [5]. Additionally, occupancy detection plays a crucial role in security systems, allowing for the timely detection of intruders and unauthorized access [6] Moreover, occupancy information can be leveraged to provide personalized experiences for occupants. Smart automation systems can adapt the behavior based on the presence or absence of occupants, offering convenience and comfort tailored to individual preferences [7]. For example, lights can be dimmed or turned off when a room is unoccupied, and temperature settings can be adjusted based on occupancy patterns throughout the day.
b. Challenges in occupancy detection: Despite the potential benefits, occupancy detection in smart environments poses several challenges. One of the primary challenges is the diversity and complexity of sensor data collected in smart environments. Sensor data streams may exhibit variability due to factors such as sensor placement, environmental conditions, and occupant behavior [8]. As a result, developing robust and accurate occupancy detection models requires handling noisy and heterogeneous data sources effectively. Another challenge is the need to balance accuracy with privacy concerns. Occupancy detection systems must respect the privacy and security of occupants, particularly in residential settings. Therefore, it is essential to design algorithms that can infer occupancy status without compromising individual privacy or collecting sensitive information [9]. Furthermore, deploying occupancy detection systems in real-world environments introduces challenges related to system integration, scalability, and reliability. Smart building automation systems often consist of heterogeneous devices and platforms, requiring seamless integration of occupancy detection algorithms within the existing infrastructure [10]. Moreover, occupancy detection systems must be scalable to accommodate large-scale deployments in diverse occupancy settings while ensuring robust performance under varying conditions.
c. State-of-the-art techniques: Various techniques have been proposed for occupancy detection in smart buildings. These techniques range from traditional, simple rule-based approaches to sophisticated, intricate, and advanced machine learning algorithms. Rule-based methods rely on predefined heuristics and thresholds to infer occupancy status from sensor data. While the technique is simple and interpretable, rule-based approaches may struggle to capture complex patterns and adapt to changing environments [11]. Machine learning techniques, particularly supervised learning algorithms, have gained popularity for occupancy detection tasks. These algorithms learn patterns from labeled training data and can generalize to unseen instances thus facilitating their application to real-world applications [12]. support vector machines (SVMs), artificial neural networks (ANNs), and random forests (RFs) are a few of the commonly used machine learning algorithms for occupancy detection.

Recent advancements in deep learning, specifically CNNs, have shown promising results in occupancy detection tasks. CNNs can automatically learn hierarchical representations of sensor data, capturing intricate patterns and correlations [13]. By leveraging the spatial and temporal characteristics of sensor data, CNN-based models can achieve higher accuracy and robustness compared to traditional machine learning approaches.

2. LITERATURE REVIEW

The literature on smart building occupancy detection encompasses a wide range of research efforts aimed at developing efficient and accurate methods for identifying occupancy status in smart environments. smart environments are designed to keep sustainability as the focal point of interest, alongside facilitating human comfort by managing services, security, and resources. This literature review is a recapitulation of the significant research, refinements, and breakthroughs in the field, emphasizing different approaches, challenges, and emerging trends.

2.1. Rule-based approaches

Early studies on occupancy detection often employed rule-based approaches, where occupancy status was inferred using predefined heuristics and thresholds. Petrosanu *et al.* [14] have compared and highlighted the work of numerous scientists to propose a rule-based method that combined inputs from motion sensors, door contacts, and light sensors to determine occupancy. While rule-based approaches are intuitive, simple to understand, and straightforward to implement, they may lack flexibility, and struggle to adapt to diverse environmental conditions. Adaptability in rule-based systems involves manual adjustment of the pre-defined rules, in response to changes in the situations; thus, making the system static, inflexible, and non-scalable.

2.2. Machine learning techniques

With the advent of machine learning, researchers began exploring supervised learning algorithms for occupancy detection tasks. Support vector machines (SVMs), decision trees (DT), k-nearest neighbors (KNN), naïve Bayes (NB), and random forests (RFs) are among the commonly used machine learning techniques in this context. For instance, Djenouri *et al.* [15] and Dai *et al.* [16] applied SVMs to sensor data collected from smart homes and achieved promising results in occupancy classification. Similarly, Koklu and Tutuncu [17] utilized DT whereas Sarker *et al.* [18] deployed deep neural networks (DNN) to infer occupancy status based on features extracted from sensor data streams.

2.3. Deep learning approaches

In recent years, deep learning techniques, particularly CNNs, have gained attraction in occupancy detection research. CNNs have shown remarkable capabilities in capturing complex patterns and spatial dependencies in sensor data. For example, Yu *et al.* [19] proposed a CNN-based approach for occupancy detection using data from motion sensors and achieved superior performance compared to traditional machine learning methods. Hung and Chakrabarti [20] and Mtibaa *et al.* [21] have developed a deep learning model based on long short-term memory (LSTM) networks for real-time occupancy prediction in parking spaces and smart buildings.

2.4. Sensor fusion techniques

Another area of interest in occupancy detection research is sensor fusion, where data from multiple sensor modalities are integrated to improve accuracy and reliability. Roselyn *et al.* [22] and Tan *et al.* [23] investigated the fusion of data from motion sensors, light sensors, and door contacts using a probabilistic framework and demonstrated enhanced performance in occupancy detection. Chan *et al.* [24] and Nesa and Banerjee [25] have proposed a sensor fusion approach based on Bayesian inference for occupancy estimation in smart offices.

3. DATASET

The UCI Occupancy Dataset provides a rich collection of sensor data, including features such as date, temperature, humidity, light, CO_2 levels, humidity ratio, and occupancy status. This dataset offers valuable insights into occupancy patterns within smart environments, facilitating the development of occupancy detection models and smart building automation systems.

a. Date: The timestamp indicates when sensor readings were recorded, providing temporal information for analyzing occupancy patterns over time.
b. Temperature: Continuous measurements of temperature captured by sensors deployed within the house.
c. Temperature fluctuations may indicate changes in occupancy or environmental conditions.
d. Humidity: The level of humidity in the air, measured by humidity sensors. Humidity variations can affect occupant comfort and indoor air quality.
e. Light: Binary values representing the presence or absence of light in different areas of the house. Light sensor readings can help infer occupancy status and activity patterns.
f. CO_2 levels: Measurements of carbon dioxide (CO_2) concentration in the air, indicating indoor air quality and occupancy-related CO_2 emissions.
g. Humidity ratio: A calculated parameter derived from temperature and humidity measurements, providing additional insights into indoor environmental conditions.
h. Occupancy: Binary labels indicating whether a particular area or room in the house is occupied or unoccupied. This target variable is crucial for training occupancy detection models.

Analyzing all these features collectively has led to the development of machine learning models to predict occupancy status based on sensor data. Additionally, the dataset enables exploratory data analysis, feature engineering, model training, and evaluation to uncover occupancy patterns, optimize energy usage, and enhance home automation systems' efficiency.

4. PROPOSED MODEL

In the context of the UCI House Occupancy Dataset, we propose a machine-learning model for occupancy detection based on the provided sensor data. The model aims to accurately predict whether a particular area within a house is occupied or unoccupied at any given time. It leveraging features such as temperature, humidity, light, CO_2 levels, humidity ratio, and timestamps.

4.1. Model architecture and principles of Neutrosophy

This model proposes a framework that integrates neutrosophic convolution, neutrosophic max-pooling, and neutrosophic logic layers into a CNN architecture. The neutrosophic convolutional layer is designed to extract features from input data while considering indeterminacy, uncertainty, and contradictory information. The neutrosophic max-pooling layer aggregates the extracted features to reduce dimensionality and enhance computational efficiency. The neutrosophic logic layer handles reasoning and decision-making processes by incorporating neutrosophic logic principles.

The CNN model architecture consists of: i) A convolutional layer with 16 kernels and a kernel size of 3×3, followed by a rectified linear unit (ReLU) activation function; ii) A max-pooling layer with a pooling size of 2×2 to downsample the feature maps; iii) A flattening layer to convert the 2D feature maps into a 1D vector; and iv) Two fully connected dense layers with 64 and 1 neurons, respectively, activated by ReLU and sigmoid functions. After compiling the model using the Adam optimizer and binary cross-entropy loss, it is trained using example input data represented as neutrosophic sets. The input data includes temperature, humidity, light, CO_2, and humidity ratio. Subsequently, the model predicts labels based on the input data. Ground truth labels (1 designating occupied state and 0 designating unoccupied state) are used to compute the evaluation metrics such as accuracy, precision, recall, and F1-score. Finally, the predicted labels and evaluation metrics are printed for analysis.

4.2. Model training

The step-by-step algorithm for training a neural network model using neutrosophic layers is shown:

a. Data preparation
   - Organize the dataset containing input features and target labels.
   - Ensure the dataset is properly formatted and cleaned.
   - Split the dataset into training and testing sets.

b. Model initialization
   - Import the necessary libraries, including neutrosophic layers, TensorFlow, and NumPy.
   - Initialize neutrosophic convolutional, max-pooling, and logic layers.
   - Define the architecture of the convolutional neural network model using TensorFlow's Keras API.
   - Specify the number of kernels, kernel size, pooling size, activation functions, and input shape.

c. Model compilation
   - Compile the CNN model by specifying: i) The optimizer (e.g., Adam, SGD) to minimize the loss function; ii) The loss function (e.g., binary cross-entropy, mean squared error) to measure the model's performance; and iii) Evaluation metrics (e.g., accuracy, precision, recall) to monitor during training.

d. Model training
   - Use the training dataset to train the model.
   - Iterate over multiple epochs (complete passes through the entire dataset) to optimize the model's parameters.
   - Specify the batch size and the number of epochs. Batch refers to the number of samples processed before updating the model and epochs refers to the iterations over the entire dataset during training.
   - During each epoch, feed batches of input data into the model and adjust the weights using backpropagation.

e. Model evaluation
   - Evaluate the trained model's performance using the testing dataset.
   - Assess the model's effectiveness by computing evaluation metrics such as accuracy, precision, recall, and F1-score.
   - Compare the predicted labels with the ground truth labels to measure the model's accuracy and generalization ability.

f. Model optimization
   - Model may be fine-tuned by adjusting different hyperparameters (e.g., learning rate, dropout rate) or exploring different architectures (e.g., adding more layers, changing activation functions).
   - Utilize techniques such as regularization, dropout, or batch normalization to prevent overfitting and improve model performance.

g. Model deployment
   - Deploy the model for real-world occupancy detection applications, once it delivers satisfactory performance.
   - Use the trained model to make predictions on new, unseen data.
   - Continuously monitor the model's performance and periodically retrain to accommodate changes in the data distribution.

This algorithm outlines the key steps involved in training a neural network model with neutrosophic layers and provides a structured approach to developing and evaluating the model.

4.3. Model working

Once trained and validated, the occupancy detection model can be deployed in real-world scenarios. Given new sensor data as input, the model computes the probability of occupancy for the target area. If the predicted probability exceeds a predefined threshold (e.g., 0.5), the model classifies the area as occupied; otherwise, it classifies it as unoccupied.

By continuously monitoring sensor data and making real-time occupancy predictions, the model can support various applications, including smart energy management, security systems, and personalized building automation. Additionally, the model's performance can be periodically evaluated and fine-tuned to make it adaptable to frequent changes in occupancy patterns and environmental conditions. A mathematical model for training a CNN for occupancy detection is shown in Algorithm 1.

Algorithm 1. Convolutional neural network for occupancy detection

Input:
Training dataset: $\{(X(i), y(i))\} i = 1\, m$ where $X(i)$ is the input feature matrix of size ($nsamples$, $nfeatures$, and $nchannels$) and $y(i)$ is the corresponding binary occupancy label vector of size ($nsamples$, 1).
Testing dataset: $\{(Xtest(i), ytest(i))\} i = 1\, m$ test for evaluation.
Output: Trained CNN model f for occupancy detection.
Algorithm:
1. Initialization:
   - Initialize the CNN architecture with neutrosophic layers:
     - Neutrosophic convolutional layer: $Convneutrosophic$
     - Neutrosophic max-pooling layer: $MaxPoolneutrosophic$
     - Neutrosophic logic layer: $Logicneutrosophic$
2. Define CNN architecture:
   - Define the architecture of the CNN model using the neutrosophic layers:
     - $Convneutrosophic$ with $nkernels$ kernels, kernel size ($kheight, kwidth$), and ReLU activation function.
     - $MaxPoolneutrosophic$ with pooling size ($pheight, pwidth$).
     - Flatten layer to convert the 2D feature maps into a 1D vector.
     - Fully connected Dense layer with n units neurons and ReLU activation function.
     - Output layer with 1 neuron and sigmoid activation function.
3. Compile the model:
     - Compile the model using the Adam optimizer.
     - Use binary cross-entropy loss function for binary classification.
     - Monitor accuracy as the evaluation metric.
4. Model training:
   - Train the model on the training dataset:
     - Feed batches of input data $X(i)$ into the model.
     - Model parameters may be optimized using backpropagation and gradient descent.
     - Repeat for multiple epochs until convergence.
5. Model evaluation:
   - Measure the performance of the trained model using the testing dataset:
     - Assess the model's effectiveness in occupancy detection by calculating accuracy, precision, recall, and F1-score.
     - Compare the predicted occupancy labels with the ground truth labels.
6. Model optimization:
   - Fine-tune the model by adjusting hyperparameters or exploring different architectures.
   - Utilize techniques such as regularization or dropout to prevent overfitting.
7. Model Deployment:
   - On satisfactory performance, deploy the model for occupancy detection applications.
   - Use the trained model to predict occupancy status based on input sensor data.
   - Monitor the model's performance and update it as needed to adapt to changes in the environment.

This algorithm outlines the step-by-step process of training a CNN with neutrosophic layers for occupancy detection, including model initialization, architecture definition, training, evaluation, optimization, and deployment.

5. RESULTS AND DISCUSSION

Variations in occupancy dataset parameters (temperature, humidity, light, CO_2 levels, humidity ratio, and occupancy) have been studied over time. Figure 1 shows the Heat map of the correlated parameters in the model. Here's a breakdown of the analysis:

a. Temperature: The temperature fluctuates slightly around 21.76 to 21.89 degrees Celsius over time
b. Humidity: Humidity ranges from 30.29% to 31.89% with minor fluctuations over time.
c. Light: Light intensity remains relatively constant, mostly around 429 to 449.5 units, with occasional fluctuations.
d. CO_2 Levels: CO_2 levels vary from 793.5 to 1110 parts per million (ppm) throughout the observation period.
e. Humidity ratio: The humidity ratio ranges from 0.004908 to 0.005185, indicating some variability but generally stable conditions.
f. Occupancy: Occupancy status is primarily occupied (1) throughout the recorded time, with the last few entries showing both occupied and unoccupied states.
   – Temperature, humidity, and CO_2 levels demonstrate minor fluctuations within relatively narrow ranges.
   – Light intensity remains fairly consistent.
   – The occupancy status is predominantly occupied, with intermittent unoccupied periods towards the end of the data.
g. Additional insights: i) Further analysis could explore correlations between different variables and their impact on occupancy status; and ii) Anomalies or patterns may emerge when analyzing the data over longer periods or comparing it with external factors such as time of day or occupancy patterns in the building.

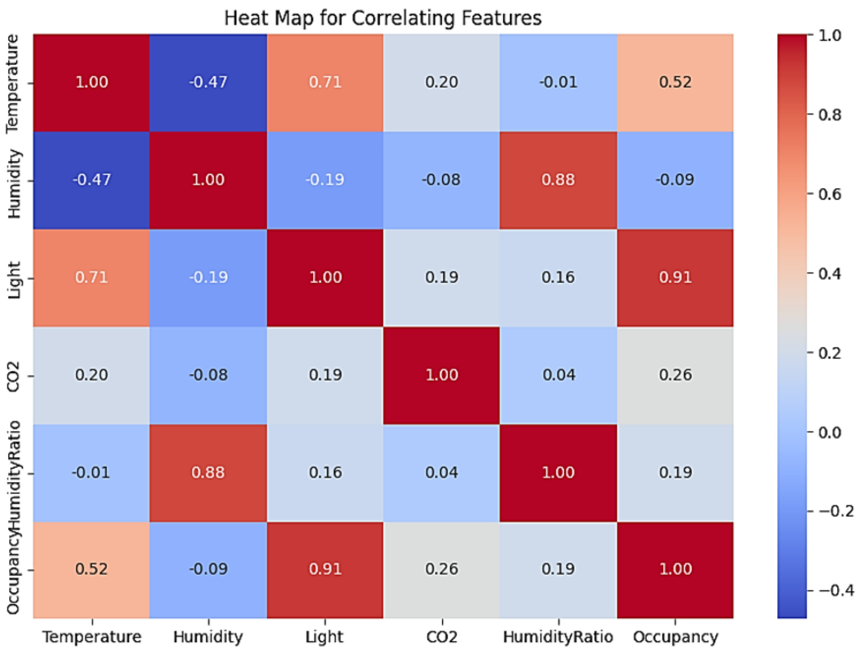

Figure 1. Correlation of features used as input to model overall trends

Figure 2 depicts the model performance in terms of accuracy versus loss. The model has achieved an impressive accuracy score of 99.18%, indicating that the integration of Neurtosophy into the CNN model correctly classifies the occupancy status in the test dataset with high precision This high accuracy suggests that the model has effectively learned from the training data and accurately predicts for unseen data. The model performs exceptionally well in both classes. For class 0 (unoccupied), it achieves a precision, recall, and F1-score of 1.00, indicating perfect performance. For class 1 (occupied), it achieves slightly lower but still excellent precision (0.97), recall (0.99), and F1-score (0.98). These metrics indicate that the model accurately identifies both unoccupied and occupied instances, with very few misclassifications. The model demonstrates robust performance in predicting occupancy status, showcasing its effectiveness in handling uncertainties within the dataset and aligning with the principles of Neutrosophy by acknowledging and navigating uncertainties within its predictions. Figure 3 shows the plot of predicted probabilities and uncertainties of the samples. This analysis provides a basic overview of the provided data. Further insights and conclusions can be drawn by applying advanced statistical techniques or domain-specific knowledge.

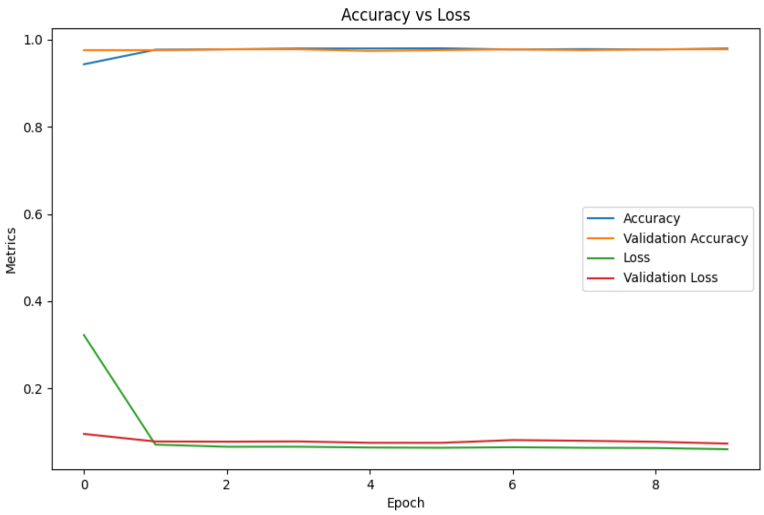

Figure 2. Model performance in terms of accuracy vs loss of the proposed model

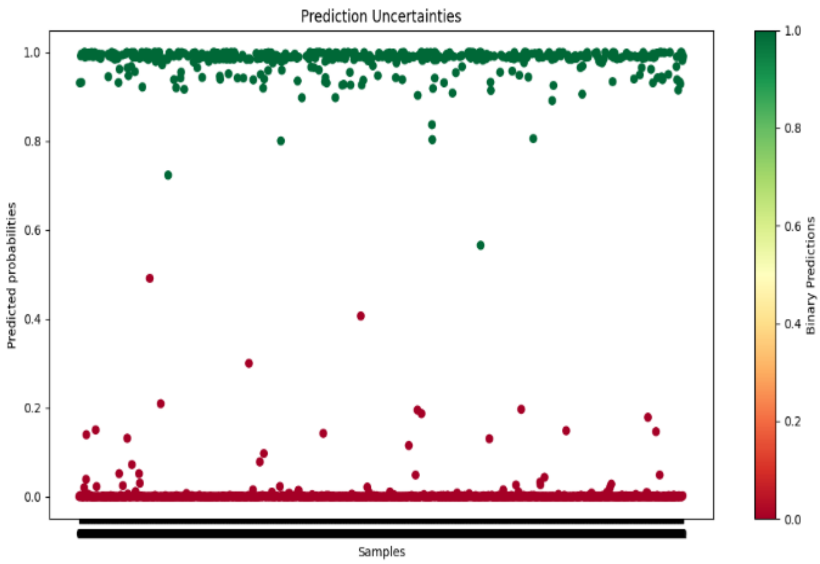

Figure 3. Plot depicting prediction uncertainties and predicted probabilities over the sample range

6. CONCLUSION

This paper proposed an improvised CNN model for occupancy detection based on sensor data. The model architecture, designed with the principles of Neutrosophy in mind, demonstrates prowess in the handling of uncertainties inherent in both the dataset and model predictions. Through the utilization of a multilayer perceptron (MLP) architecture, the constructed model is capable of effectively handling tabular data and binary classification tasks. The model comprises input, hidden, and output layers, each meticulously crafted to navigate and encapsulate uncertainties within the data and predictions.

During training and evaluation, the model exhibited exceptional performance, achieving an accuracy score of approximately 99.18%. This high accuracy underscores that the model can generalize well and make faultless predictions on unknown test data, thus instilling confidence in its predictive capabilities. Further analysis of the classification report reveals that the model performs exceptionally well in identifying both occupied and unoccupied instances. With precision, recall, and F1-score metrics ranging from 0.97 to 1.00, the model demonstrates a robust understanding of the underlying patterns and relationships in the data. In conclusion, the Neutrosophic-enhanced CNN model not only provides accurate occupancy predictions but also navigates and embraces uncertainties inherent in the dataset and model predictions. By aligning with the principles of Neutrosophy, the model serves as a testament to the importance of acknowledging and exploring uncertainties in the pursuit of robust and reliable predictive models.

REFERENCES

[1] A. Singh, V. Kansal, M. Gaur, and M. S. Pandey, "Predicting smart building occupancy using machine learning," *Lecture Notes in Networks and Systems*, vol. 479, pp. 145–151, 2023, doi: 10.1007/978-981-19-3148-2_12.

[2] M. Esrafilian-Najafabadi and F. Haghighat, "Occupancy-based HVAC control systems in buildings: a state-of-the-art review," *Building and Environment*, vol. 197, 2021, doi: 10.1016/j.buildenv.2021.107810.

[3] H. Yar, A. S. Imran, Z. A. Khan, M. Sajjad, and Z. Kastrati, "Towards smart home automation using IoT-enabled edge-computing paradigm," *Sensors*, vol. 21, no. 14, 2021, doi: 10.3390/s21144932.

[4] N. Somu, G. Raman M R, and K. Ramamritham, "A deep learning framework for building energy consumption forecast," *Renewable and Sustainable Energy Reviews*, vol. 137, 2021, doi: 10.1016/j.rser.2020.110591.

[5] N. S. Fayed, M. M. Elmogy, A. Atwan, and E. El-Daydamony, "Efficient occupancy detection system based on neutrosophic weighted sensors data fusion," *IEEE Access*, vol. 10, pp. 13400–13427, 2022, doi: 10.1109/ACCESS.2022.3146346.

[6] Z. Chen, C. Jiang, M. K. Masood, Y. C. Soh, M. Wu, and X. Li, "Deep learning for building occupancy estimation using environmental sensors," *Studies in Computational Intelligence*, vol. 865, pp. 335–357, 2020, doi: 10.1007/978-3-030-31760-7_11.

[7] G. Gao, J. Li, and Y. Wen, "Energy-efficient thermal comfort control in smart buildings via deep reinforcement learning," *arXiv preprint arXiv:1901.04693*, 2019.

[8] L. Rueda, K. Agbossou, A. Cardenas, N. Henao, and S. Kelouwani, "A comprehensive review of approaches to building occupancy detection," *Building and Environment*, vol. 180, 2020, doi: 10.1016/j.buildenv.2020.106966.

[9] H. Wang, J. Zhang, C. Lu, and C. Wu, "Privacy preserving in non-intrusive load monitoring: a differential privacy perspective," *IEEE Transactions on Smart Grid*, vol. 12, no. 3, pp. 2529–2543, 2021, doi: 10.1109/TSG.2020.3038757.

[10] A. Soleimanijavid, I. Konstantzos, and X. Liu, "Challenges and opportunities of occupant-centric building controls in real-world implementation: a critical review," *Energy and Buildings*, vol. 308, 2024, doi: 10.1016/j.enbuild.2024.113958.

[11] M. S. Aliero *et al.*, "Non-intrusive room occupancy prediction performance analysis using different machine learning techniques," *Energies*, vol. 15, no. 23, 2022, doi: 10.3390/en15239231.

[12] P. W. Tien, S. Wei, J. Darkwa, C. Wood, and J. K. Calautit, "Machine learning and deep learning methods for enhancing building energy efficiency and indoor environmental quality–a review," *Energy and AI*, vol. 10, 2022, doi: 10.1016/j.egyai.2022.100198.

[13] C. Feng, A. Mehmani, and J. Zhang, "Deep learning-based real-time building occupancy detection using AMI data," *IEEE Transactions on Smart Grid*, vol. 11, no. 5, pp. 4490–4501, 2020, doi: 10.1109/TSG.2020.2982351.

[14] D. M. Petrosanu, G. Carutasu, N. L. Carutasu, and A. Pîrjan, "A review of the recent developments in integrating machine learning models with sensor devices in the smart buildings sector with a view to attaining enhanced sensing, energy efficiency, and optimal building management," *Energies*, vol. 12, no. 24, 2019, doi: 10.3390/en12244745.

[15] D. Djenouri, R. Laidi, Y. Djenouri, and I. Balasingham, "Machine learning for smart building applications: review and taxonomy," *ACM Computing Surveys*, vol. 52, no. 2, 2019, doi: 10.1145/3311950.

[16] X. Dai, J. Liu, and X. Zhang, "A review of studies applying machine learning models to predict occupancy and window-opening behaviours in smart buildings," *Energy and Buildings*, vol. 223, 2020, doi: 10.1016/j.enbuild.2020.110159.

[17] M. Koklu and K. Tutuncu, "Tree based classification methods for occupancy detection," in *IOP Conference Series: Materials Science and Engineering*, 2019, vol. 675, no. 1, doi: 10.1088/1757-899X/675/1/012032.

[18] A. Sarker *et al.*, "Deep learning based prediction towards designing a smart building assistant system," in *Proceedings - 2020 IEEE 17th International Conference on Mobile Ad Hoc and Smart Systems, MASS 2020*, 2020, pp. 202–210, doi: 10.1109/MASS50613.2020.00034.

[19] J. Yu, A. de Antonio, and E. Villalba-Mora, "Deep learning (CNN, RNN) applications for smart homes: a systematic review," *Computers*, vol. 11, no. 2, 2022, doi: 10.3390/computers11020026.

[20] B. T. Hung and P. Chakrabarti, "Parking IoT occupancy detection using hybrid deep learning CNN-LSTM approach," in *Proceedings of 2nd International Conference on Artificial Intelligence: Advances and Applications (ICAIAA 2021)*, 2022, pp. 501–509, doi: 10.1007/978-981-16-6332-1_43.

[21] F. Mtibaa, K. K. Nguyen, M. Azam, A. Papachristou, J. S. Venne, and M. Cheriet, "LSTM-based indoor air temperature prediction framework for HVAC systems in smart buildings," *Neural Computing and Applications*, vol. 32, no. 23, pp. 17569–17585, 2020, doi: 10.1007/s00521-020-04926-3.

[22] J. P. Roselyn, R. A. Uthra, A. Raj, D. Devaraj, P. Bharadwaj, and S. V. D. Krishna Kaki, "Development and implementation of novel sensor fusion algorithm for occupancy detection and automation in energy efficient buildings," *Sustainable Cities and Society*, vol. 44, pp. 85–98, 2019, doi: 10.1016/j.scs.2018.09.031.

[23] S. Y. Tan, M. Jacoby, H. Saha, A. Florita, G. Henze, and S. Sarkar, "Multimodal sensor fusion framework for residential building occupancy detection," *Energy and Buildings*, vol. 258, 2022, doi: 10.1016/j.enbuild.2021.111828.

[24] Z. Chen, C. Jiang, and L. Xie, "Building occupancy estimation and detection: a review," *Energy and Buildings*, vol. 169, pp. 260–270, 2018, doi: 10.1016/j.enbuild.2018.03.084.

[25] N. Nesa and I. Banerjee, "IoT-based sensor data fusion for occupancy sensing using Dempster-Shafer evidence theory for smart buildings," *IEEE Internet of Things Journal*, vol. 4, no. 5, pp. 1563–1570, 2017, doi: 10.1109/JIOT.2017.2723424.

BIOGRAPHIES OF AUTHORS

Ranjeeta Mittal received the B.E. degree in computer science and engineering from Deenbandhu Chhotu Ram University of Science and Technology, Murthal, India in 1998. She received the M.E. in computer science and engineering from the National Institute of Technical Teachers Training and Research, Chandigarh, India in 2009. Currently, she is an assistant professor at the Department of Computer Science and Engineering, Manav Rachna International Institute of Research and Studies, Faridabad, India. Her research interests include artificial intelligence and energy conservation. She can be contacted at email: ranjeetamittal@gmail.com.

Suresh Kumar is a distinguished academician and professional associated with the Manav Rachna International Institute of Research and Studies (MRIIRS). He is a life member of the Indian Society of Technical Education (ISTE) and Computer Society India (CSI). He is also a Senior member of IEEE, USA, and IACSIT, Singapore. He has published more than eighty research papers in International Journals and Conferences. The HelixSmartLabs a startup incubated at Manav Rachna under his supervision and is registered as a Pvt. Ltd company. He has published two patents and a technology transfer in a rules package of R data analytics software (included in version 1.5.5 onwards). The package is available at https://cran.r-project.org/package=arules. He can be contacted at email: enthuvs@gmail.com.

Urvashi Chugh is currently working as an associate professor in the KIET group of Institutions, Ghaziabad, India. She received her PhD from the Jay Pee University of Information and Technology, Waknaghat, India. She did M.Tech. and BTech from MDU and Kurukshetra University respectively. Her areas of interest are sensor networks, nano networks, and molecular nano communications. She has various patents, and research papers in conferences and journals. She can be contacted at email: urvashichugh.fet@mriu.edu.in.