Hindawi
Complexity
Volume 2022, Article ID 4989344, 19 pages
https://doi.org/10.1155/2022/4989344

Research Article

CIMA: A Novel Classification-Integrated Moving Average Model for Smart Lighting Intelligent Control Based on Human Presence

Aji Gautama Putrada,[1] **Maman Abdurohman**,[2] **Doan Perdana**,[1] **and Hilal Hudan Nuha**[2]

[1]*Advanced and Creative Networks Research Center, Telkom University, Bandung, Indonesia*
[2]*School of Computing, Telkom University, Bandung, Indonesia*

Correspondence should be addressed to Maman Abdurohman; abdurohman@telkomuniversity.ac.id

Received 17 March 2022; Accepted 20 August 2022; Published 21 September 2022

Academic Editor: Gonzalo Farias

Smart lighting systems utilize advanced data, control, and communication technologies and allow users to control lights in new ways. However, achieving user comfort, which should be the focus of smart lighting research, is challenging. One cause is the passive infrared (PIR) sensor that inaccurately detects human presence to control artificial lighting. We propose a novel classification-integrated moving average (CIMA) model method to solve the problem. The moving average (MA) increases the Pearson correlation (PC) coefficient of motion sensor features to human presence. The classification model is for a smart lighting intelligent control based on these features. Several classification models are proposed and compared, namely, *k*-nearest neighbor (KNN), support vector machine (SVM), decision tree (DT), näive Bayes (NB), and ensemble voting (EV). We build an Internet of things (IoT) system to collect movement data. It consists of a PIR sensor, a NodeMCU microcontroller, a Raspberry Pi-based platform, a relay, and LED lighting. With a sampling rate of 10 seconds and a collection period of 7 days, the system achieved 56852 data records. In the PC test, movement data from the PIR sensor has a correlation coefficient of 0.36 to attendance, while the MA correlation to attendance can reach 0.56. In an exhaustive search of an optimum classification model, KNN has the best and the most robust performance, with an accuracy of 99.8%. It is more accurate than direct light control decisions based on motion sensors, which are 67.6%. Our proposed method can increase the correlation value of movement features on attendance. At the same time, an accurate and robust KNN classification model is applicable for human presence-based smart lighting control.

1. Introduction

Smart lighting systems utilize advanced data, control, and communication technologies and allow users to control lights in new ways [1]. Smart lighting products are already on the market, where their global revenue is up to US$600 million in 2020 [2]. The main issue of smart lighting research is energy efficiency, in which until 2021, 232 out of 384 papers on smart lighting try to solve this problem [3]. The main targets for smart lighting installations are on roads, offices, and housings [4]. Noting the needs of such targets, user comfort and security also become important in smart lighting. However, achieving user comfort is still challenging because the passive infrared (PIR) sensor, a low-price movement sensor, inaccurately detects human presence to control artificial lighting [5].

A smart thing device such as smart lighting should be able to co-operate with its users and environment intelligently [6]. Gartner stated that intelligence is one of five key factors in smart lighting. Activity recognition is an example of intelligence implementation, where it detects human activity based on machine learning applications on several types of sensors [7]. Intelligence can also be applied to improve uncertainty problems in conventional control systems, hence creating an intelligent control system [8].

Several previous studies have tried to overcome the problem of motion sensors to improve accuracy for smart lighting intelligent control based on human presence. Jin

et al. [9] used a time-series-artificial neural network (TS-ANN) on historical PIR sensor data and got up to 97% accuracy in human presence predictive control based on human presence. Fakhruddin et al. [10] used activity recognition to detect five activities using four PIR sensors installed in the house using the principal component analysis-k-nearest neighbor (PCA-KNN) method and to get an accuracy of 94%. Lupion et al. [11] made another study that uses activity recognition and utilizes feature extraction from sliding windows on various sensor data used to produce 99.26% accuracy in detecting 14 activities using the random forest classification method. Park et al. [12] used reinforcement learning (RL) on the PIR sensor and several other sensors to get smart lighting that is adaptive to user needs and also energy-efficiency.

Reconsidering [9, 11], we can think of human presence as a type of activity. On the other hand, we can also consider historical data as a sliding window feature extraction. A moving average (MA) concept can substitute the sliding window feature extraction method in this intuition. Usually, MA is a method for smooth fluctuating data and, among others, can be used as a noise filtering method for time-series data [13]. In some research, MA is used to increase the Pearson correlation (PC) coefficient of machine learning features [14]. Furthermore, we can conduct a comprehensive test to find the optimum classification model. Several studies use some well-known classical machine learning methods such as KNN, support vector machine (SVM), decision tree (DT), and näive Bayes (NB) to train the classification model [15]. Other research also uses ensemble learning methods such as ensemble voting (EV) to improve the performance of the existing classical machine learning method [16].

We propose a novel classification-integrated MA (CIMA) model method to solve the problem. The MA is to increase the correlation of motion sensor features to human presence, while the classification model is for a smart lighting intelligent control based on these features. We train the proposed classification model with KNN, SVM, DT, and NB. We also use ensemble learning methods such as EV to improve classical machine learning performance. An Internet of things (IoT) system is built on a test-bed environment to retrieve movement data from the PIR sensor. At the end-device layer, the microcontroller used is NodeMCU. We build a Node-Red server on the Raspberry Pi at the Platform layer. It stores the movement data log in a comma-separated value (CSV) file. We use test parameters such as accuracy, precision, recall, and $F1$-score to discover the optimal classification model. In addition, to check the robustness of the model, the cross-validation method is used.

The main contributions of our work are listed below:

Increasing the correlation between movement data and human presence through the MA method. A novel classification model with significant accuracy from state-of-the-art research utilizing MA data from movement data as a feature. An accurate yet low-price solution for human presence-based smart lighting control because of the utilization of motion sensors.

The remainder of this document has the following systematic: Section 2 presents works related to the research undertaken, Section 3 describes methods used in this research, Section 4 gives the results of the tests conducted, Section 5 reports the results and compares them with state-of-the-art studies while highlighting the contributions provided from our work, and finally, Section 6 emphasizes the important findings of this study.

2. Related Works

Several studies have discussed automatic smart lighting control using the PIR sensor. Jin et al. [9] aimed to improve the accuracy of PIR sensors using the time-series-artificial neural network (TS-ANN) method and compared several features such as time, occupied ratio, time steps, and historical occupied state data. The study showed that the proposed method can provide up to 97% accuracy for the intelligent control. Putrada et al. [17] used a hierarchical hidden Markov model (HHMM) to classify five different types of activities from four PIR sensors to control smart lighting in offices. The HHMM model tested is better than the hidden Markov model (HMM), NB, and KNN method and has an accuracy of 87.6%. Ramadhan et al. [5] also used the HHMM method on 14 different activities from five PIR sensors. The accuracy of HHMM was 93%, and the method was superior to HMM. Fakhruddin et al. [10] used activity recognition to detect five activities using four PIR sensors installed in the house using the principal component analysis-k-nearest neighbor (PCA-KNN) method and to get an accuracy of 94%. Each study investigates a different amount of activity and obtains varying performance. There is an opportunity to find a correlation between the number of activities and performance in using PIR sensors for activity recognition on smart lighting. Other factors are also opportunities for investigation.

Furthermore, other studies also conducted smart lighting control but with devices or sensors other than the PIR sensor. Dai et al. [18] used five low-resolution cameras to detect nine activities in a smart lighting environment. The study provided a solution that ensures privacy even when using a camera, while the accuracy is up to 89.6%. Chun et al. [19] used a depth camera to detect four human activities in a room. The proposed method results provided 100% accuracy for the location where people are and 78.3% accuracy for the type of performed activity. Lupion et al. [11] used PIR sensors and also several different sensors including smartwatches and real-time location systems. The research produced 99.26% accuracy in detecting 14 activities using a random forest classification method. Park et al. [12] used a light sensor and actuators such as Switchmate. Switchmate consists of a motor and a position sensor to control and monitor a conventional light switch. The average light utility ratio (LUR) of the research is 67%. The studies mentioned have performance that vary from inadequate to highly adequate results. However, the equipment used is expensive when compared to the PIR sensor, which costs around US$ 1. There is an opportunity to implement an accurate and low-priced solution using CIMA and PIR sensors.

Several previous studies have applied MAs for smoothing and increasing the PC between two variables.

Husnayain et al. [20] used MA to increase the correlation between the incidence of dengue fever with Google search activities for dengue and found that the correlation was very high between the two. Hu et al. [21] utilized MAs to reduce noise in water pH and water temperature data to improve the correlation of the two data with other water quality data to provide better performance in mariculture water quality forecasting. Peng et al. [22] used MA to increase the PC between drought and flood to predict the occurrence of these two disasters in China. Badr et al. [23] showed that the correlation between the mobility ratio and growth rate ratio increased as the MA window size increased but slowly decreased when the window size was too large. Singh et al. [24] used MAs to refine CO_2 sensor readings and improved the correlation of sensor data with respiratory rate and Hjorth activity in a cardiorespiratory assessment. The results of the mentioned studies show that there is an opportunity to apply MA to movement data to increase the PC coefficient for an accurate classification model.

3. Materials and Methods

3.1. Research Methodology. This section discusses the research methodology, from how the test data were collected, to how we obtained the final model. The methodology for developing a classification model to predict human presence is shown in Figure 1.

The PIR sensor is one of the most utilized sensor in smart lighting control [25]. We build an IoT system with PIR sensors to collect human movement data. Labeling is done to each movement as to whether there are people or not at each given moment. The system stores the data in a CSV file for further analysis. The next step is to apply the MA and observe the PC coefficient. Further is to prepare data before conducting classification training with methods KNN, SVM, DT, and NB. The possibility of applying EV to improve the performance of the classification model is analyzed later. The last step is to analyze the most optimum model and perform cross-validation to check for possible overfitting.

3.2. Smart Lighting IoT System. The IoT architecture of the smart lighting system for automatic light control based on human presence is as shown in Figure 2. We chose a living room as a test-bed environment to implement the proposed architecture.

In the proposed IoT architecture, there are three main layers, namely, the end-device layer, platform layer, and application layer [26]. Then, there are additional communication protocols and gateways that connect the three layers. At the end-device layer, the layer directly related to the IoT hardware, the three main devices are PIR sensors, NodeMCU, and relays. The PIR sensor functions to detect human movement [27]. NodeMCU has a system on chip (SoC), ESP8266, which includes a microcontroller and WiFi communication [28]. WiFi is used for communication between the end-device layer and platform layer [29]. The relay is an actuator connected to the LED light [30]. Its function is to turn the LED on and off like a switch controlled via the microcontroller.

We build the platform layer on a Raspberry Pi (Raspi), an open-source mini-personal computer (mini-PC) running with a Raspbian operating system (OS) [31]. We use Node-Red for web service functions. Node-Red is also an open-source web service based on Node.js, which has a special add-on for IoT systems [32]. The Node-Red performs movement sensor data log dumps to a CSV file used for training the classification model. Raspi can also be used to run Python functions [33]. Hence, the classification model running in Python can be executed on this server.

The application layer is concerned with the interaction between the system and the user. Users can use the Python-based graphical user interface (GUI) to set the light status manually or automatically. Especially for testing, the user can also choose to control lights with the novel method or the conventional method, which compares the comfort between the new system and the legacy system. The platform layer links to the application and end-device layers via the Internet and the hypertext transfer protocol (HTTP) application programming interface (API) protocol.

The device is a single set and detects the presence of one person at one location in one room. A chart depicting the placement of devices in a room is shown in Figure 3. The PIR sensor, NodeMCU, and relay are on the ceiling as part of the end-device. The PIR sensor is placed approximately above where humans conduct activities, for example, working. The end devices, especially the relay, are connected to the LED light. The LED light is on the ceiling in the middle of the room. The NodeMCU receives motion sensor data, controls the LED light via relays, and communicates with the IoT Platform via WiFi. A wall-mounted WiFi-4G router connects the WiFi network with the Internet.

The motion detection distance from the sensor is 10 meters forward. In addition, the PIR sensor has a capture range as wide as 110°. Figure 4 shows the coverage area of the PIR sensor when placed on the ceiling. If, for example, the room's height is 2.4 m, with the range described previously, then the coverage area will form a cone with a base diameter of 5 m and a base radius of 2.5 m. Hence, the area of the cone base is approximately 20 m^2. The proposed smart lighting system hypothetically considers a person present if the person is in that mentioned space.

3.3. Moving Average. As the name suggests, MA is a method of averaging on time-series data in which a certain period of data (called data points) is averaged continuously and moves along the data series [34]. The data points are notated as N. Applying the MA results in a smoother data series [35]. Due to this nature, scientists and analysts utilize MAs in cases involving fluctuating data such as financial data, stock predictions, and signal filters [36, 37]. The MA formula for N values is as follows:

$$MA(n) = \frac{1}{N} \sum_{i=n-N+1}^{n} p_i, \tag{1}$$

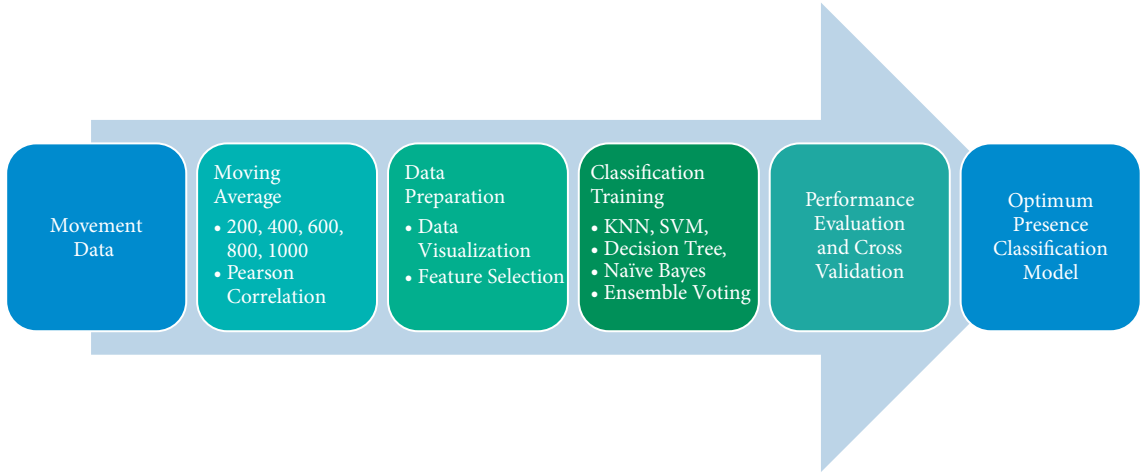

Figure 1: A chart explaining the proposed methodology for developing a classification model for predicting human presence.

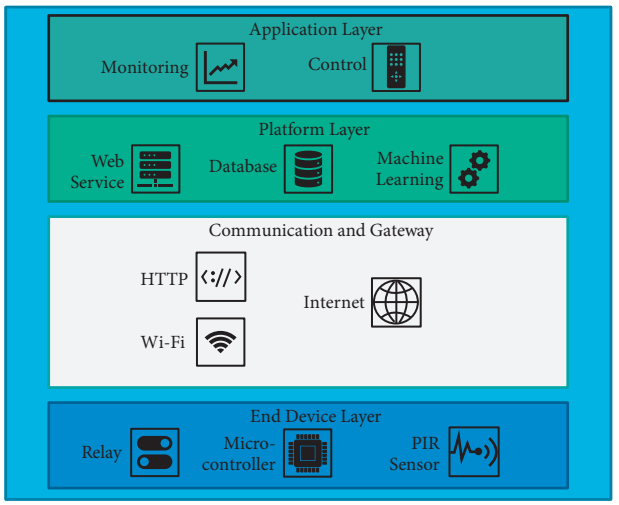

Figure 2: The IoT architecture of a smart lighting system for automatic light control based on human presence.

where p_i is the i^{th} data series in range $n-N+1$ to n and $MA(n)$ is the MA on p_n. The MA for N values and the following n data $(n+1)$ can use the following formula:

$$MA(n+1) = MA(n) + \frac{1}{N}(p_{n+1} - p_{n-N+1}). \tag{2}$$

We also introduce a novel theorem for $MA(n-1)$. The description is given in Theorem 1. This theorem is a low-level solution that simplifies the complexity of our real-time system in calculating the MA. It considers the property of the data structure in use.

Theorem 1. *If $MA(n+1)$ is given by equation (2), then $MA(n-1)$ is given by the following formula:*

$$MA(n-1) = MA(n) + \frac{1}{N}(p_{n-N} - p_n). \tag{3}$$

Proof. Consider a signal p, the $MA(n+1)$ is given by equation (2). Suppose $n = m-1$, substituting n with $m-1$ in equation (2) yields the following formula:

$$MA(m) = MA(m-1) + \frac{1}{N}(p_m - p_{m-N}). \tag{4}$$

Then, the formulas yield

$$\begin{aligned} MA(m) - \frac{1}{N}(p_m - p_{m-N}) &= MA(m-1), \\ MA(m) + \frac{1}{N}(p_{m-N} - p_m) &= MA(m-1). \end{aligned} \tag{5}$$

Moving term $MA(m-1)$ to the left side of the equation and substituting back m with n yield,

$$MA(n-1) = MA(n) + \frac{1}{N}(p_{n-N} - p_n). \tag{6}$$

□

3.4. Classification Models. Assuming our hypothesis on MA is correct, we carry out a comprehensive test to find the optimum classification model in determining attendance based on the novel movement data. The classification methods used are KNN, SVM, DT, and NB. The ensemble learning method can also improve the performance of conventional classification methods. Here we propose EV to combine several classical classification models.

KNN is a type of supervised machine learning that makes decisions based on the closest k training example to a data whose class is unknown [38]. One way to measure the closest distance of data with a training dataset is the Euclidean distance. The formula for calculating the distance in KNN with Euclidean distance is as follows:

$$\text{Distance}(x, y) = \sqrt{\sum_{k=1}^{n}(x_k - y_k)^2}, \tag{7}$$

where x is the training dataset, y is the classified data, and n is the number of features in the dataset.

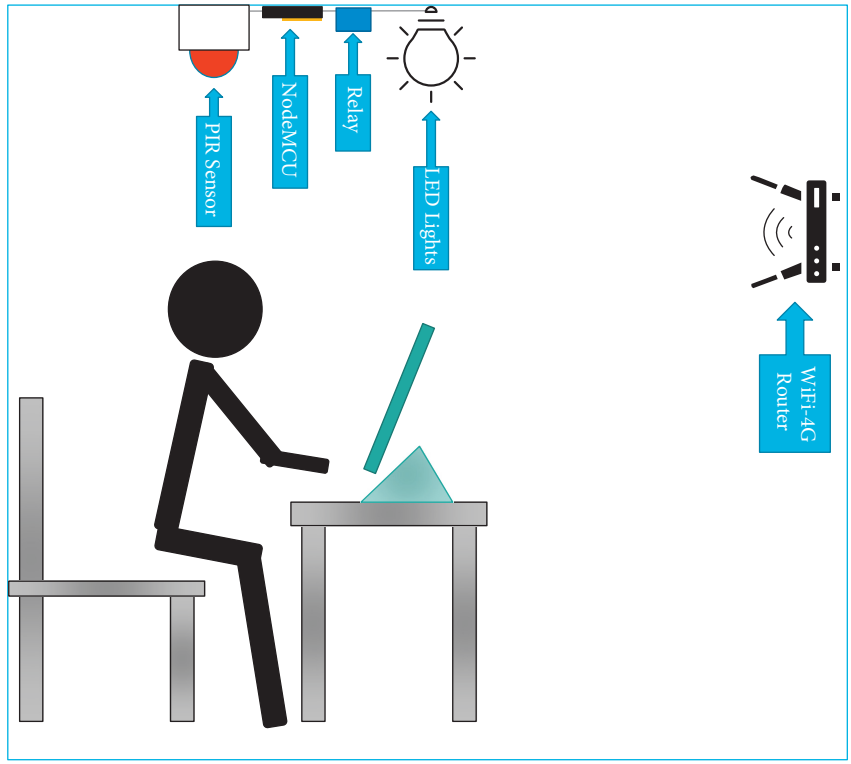

Figure 3: A chart depicting the placement of devices in a room.

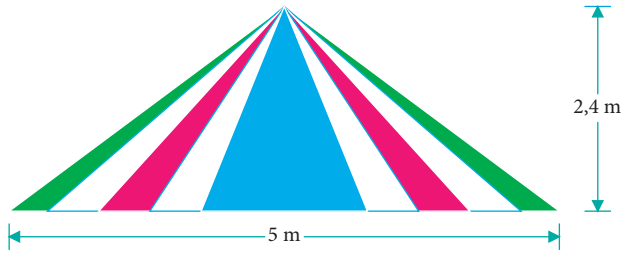

Figure 4: The coverage area of the PIR sensor if placed on the ceiling of the room.

A data structure contains the distance of y with all training examples x. As much as k training example xs closest to y are moved to a new data structure. From the k training example xs, the algorithm chooses the class with the most training example x (calculated with a mode function) as the class of y. Varying the k value influences the KNN model performance. Hence, a further test finds the optimum k value.

SVM is an example of supervised machine learning that uses margins to classify [39]. The classification method is to create a hyperplane to separate the different classes in the dataset [40]. Several kernels determine which hyperplanes can be created, including polynomial, radial basis function (RBF), and sigmoid. Polynomial kernels can use up to some different degrees. Linear kernel is considered a first-degree polynomial kernel. Here is the formula for the SVM polynomial kernel with d-degrees, including the linear kernel,

$$K(x, x') = (x.x' + r)^d, \tag{8}$$

where x and x' are vectors in the input space and r is a free parameter.

The RBF kernel is one of the most used kernel [41]. The kernel's formula of the two vectors x and x' is as follows:

$$K(x, x') = \exp\left(-\frac{\|x - x'\|^2}{2\sigma^2}\right), \tag{9}$$

where $\|x - x'\|^2$ calculates the squared Euclidean distance between x and x' and σ is a free parameter [42].

The sigmoid kernel formula for the two vectors x and x' is as follows:

$$K(x, x') = \tanh\left(\gamma x^T x' + c\right), \tag{10}$$

where γ is a free parameter with a value greater than 0 and c is a free parameter with a value less than 0.

If the dataset is linearly separable, then the suitable kernel is a linear kernel. However, if the dataset is non-linearly separable, a kernel that fits between polynomials (several d-degrees are useable), RBF, or sigmoid is the solution.

The SVM classification function is as follows:

$$f(x) = \sum_{i=0}^{n} a_i y_i K(x, x') + b, \tag{11}$$

where a_i is the Lagrange multiplier, y_i is the y value of x_i, and b is the intercept.

The DT is a classification model which is essentially a binary tree, where each branch in the tree is an ordinary if-else decision [43]. However, the if-else decision comes from a training process through several stages [44]. The two most common types of DTs are iterative dichotomiser 3 (ID3), and classification and regression tree (CART) [45]. The main difference between the two is that ID3 can only be used for classification, while CART can be used for classification as well as regression [46]. The CART formation uses a calculation of the Gini index of each feature. The Gini index describes the inequality value of a feature [47]. The lower the Gini index value, the better the feature is used to make decisions. The Gini index formula is as follows:

$$\text{Gini}(p) = 1 - \sum_{i=1}^{J} p_i^2, \tag{12}$$

where p is the feature index, p_i is the fraction of the feature p with the label i, and J is the number of labels present.

If, after the decision, the resulting class is still not uniform, then the process of calculating the Gini index for that branch is repeated for other features. The process is iterative until all branches produce a uniform class or have reached the max depth limit. Max depth is the farthest distance from the root to the leaf. Limiting the max depth value is usually to prevent overfitting.

NB classifies with the concept of the Bayes theorem, which is looking for opportunities from a hypothesis on events that have never happened [48]. NB is an efficient algorithm because each variable can be independent. The following is the formula used for the classification of NB:

$$P(c|x) = \frac{P(x|c)P(c)}{P(x)}, \tag{13}$$

where x is the data to be classified, c is the hypothetical data of a class, and $P(c|x)$ is the a posteriori probability of the data c against x.

Ensemble learning is a method of combining several learning models where the results are usually better than if only one of its members is used [49]. The downside of ensemble learning is that the algorithm is usually more computationally heavy [50]. EV is a type of ensemble learning in which, by utilizing several models from several different methods, EV selects the answer with the most number of results from each model [51]. EV can exploit the peculiarity of each member's classification model so that the advantages of each model can be seen in the results of the ensemble [52]. In hard EV, the formula used is as follows:

$$\hat{z} = \text{mode}\{X_1(y), X_2(y), \ldots, X_a(y)\}, \tag{14}$$

where $\hat{z}$ is the classification result of the EV, X is each classification model, a is the number of classification models used, and y is the data to be classified.

3.5. Evaluation Metrics. PC measures the linear correlation between two datasets [53]. The usual denotation for PC is the letter r, and the PC formula between data x and data y is as follows:

$$r = \frac{n\sum xy - (\sum x)(\sum y)}{\sqrt{\left(n\sum x^2 - (\sum x)^2\right)\left(n\sum y^2 - (\sum y)^2\right)}}, \tag{15}$$

where n is the number of records in the dataset [54].

The range of the calculated values for the PC formula is -1 to 1. There are several interpretations of the results of the PC. A negative result means that the x and y datasets have a negative correlation, where if the results are positive, the x and y data have a positive correlation. If $0.5 < |r| < 1.0$, then there is moderate to strong correlation between x and y. If $0.0 < |r| < 0.5$, there is no correlation, there is a non-linear correlation, or there is a low correlation between x and y [55].

PC is useful for feature selection in machine learning. Features with a moderate to strong correlation with the label usually pass the selection and continue to the training stage of machine learning. Features that have no correlation or low correlation are eliminated and cannot continue to the training stage [56].

The confusion matrix forms a quadrant for models with binary classification, which only involves two output values. In that quadrant, each row has data with actual positive output and data with actual negative output. Further on, each column has data with predicted positive output and data with predicted negative output. Each cell in the quadrant is an intersection between the sets of each row and each column, resulting in four possible outcomes: True Positive (TP), False Negative (FN), True Negative (TN), or False Positive (FP). The confusion matrix results show a model's predictive ability and strengthen the explanation of its accuracy, precision, recall, and $F1$-score result.

Accuracy is the ability of a model to predict data correctly. The accuracy formula is as follows:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}. \tag{16}$$

Accuracy can only measure the ability of a model to predict the correct data but cannot describe the specific capabilities of a model in making predictions. Therefore, other metrics such as precision, recall, and $F1$-score are used.

Precision shows the ability of a model to sort the negative class from the positive class. The precision formula is as follows:

$$\text{Precision} = \frac{TP}{TP + FP}. \tag{17}$$

Recall shows the ability of a model to predict the positive class. In some cases, accuracy is often mistaken for recall, whereas in imbalanced data, recall gives a true picture of the model's ability to predict positive classes. The recall formula is as follows:

$$\text{Recall} = \frac{TP}{TP + FN}. \tag{18}$$

$F1$-score is a value that describes a combination of precision and recall capabilities. The $F1$-score is different from the average because the $F1$-score uses the concept of a harmonic average. Even though it combines precision and

recall, the $F1$-score value is usually different from accuracy. The $F1-\text{score}$ formula is as follows:

$$F1-\text{score}=2\cdot\frac{\text{Precision}\times\text{Recall}}{\text{Precision}+\text{Recall}}. \tag{19}$$

Sometimes a model can experience overfitting, which is a condition when the model produces good performance on training but poor performance on validation [57]. The characteristic of an overfitting model is that it has high variance and low bias [58]. High complexity is another nature of an overfitting model. The cross-validation method can examine models with high complexity. In K-fold cross-validation, the method divides training data into several random subsamples of the same size. The fold is the term for each subsample, where K is the number of subsamples. After division, the method performs K iterations. It uses one different fold as validation data in each iteration and the rest as train data. In each iteration, accuracy or one other performance metric evaluates the model. At the end of execution, the average accuracy of each iteration becomes the final result of the cross-validation evaluation [59]. The complete process of K-fold cross-validation is given in Algorithm 1.

4. Results

4.1. IoT Implementation and Data Collection. With the IoT architecture as described in Subsection 3.2, we implement the proposed human presence-based smart lighting control. Parts of the implementation are shown in Figure 5. The main parts of the implementation are PIR sensors, NodeMCU, 4G-WiFi router, Raspberry Pi, Google Sheets, LED lighting, and relays. The Google Sheets monitors the sensed movement data. Moreover, the Raspberry Pi saves a CSV file containing movement data.

Data collection begins after the smart lighting system with the PIR sensor is successfully implemented. Movement data is collected with a sampling rate of 10 seconds and collected for seven days in one test area. During that period, the system collected 56852 data records. The data consists of movement data with binary values. A value of 1 means the PIR sensor detects movement. Otherwise, 0 means there is no movement. Each data is labeled manually. The label describes the presence of people in the room. The manual filling is done based on the presence of a subject in the room. The specification of movement data collection is given in Table 1.

A line plot can visualize how sensor data capture human movement and how the data looks compared to the actual human presence in the room. A partial snippet of the movement dataset with attendance labels is shown in Figure 6. The snippet shows data from two days out of seven days of data collection. The line plot explains that the PIR sensor reports 0 even though a subject is present. It is not that the PIR sensor is not accurate enough, but more because, while PIR sensors can only detect movement, one can imagine that subjects are not always moving while they are present. It is conceivable that if a smart lighting system directly uses the PIR sensor results for light control, the lights will turn on and off while people are still present. It results in disturbance to people's comfort.

4.2. Moving Average Application. The intuition is that the application of MA to the movement data results in a curve with a PC coefficient closer to human presence than movement data. Visualization in the form of a line plot can help illustrate this intuition. The line plot of MA results, movement data, and human presence are shown in Figure 7. Movement data of Day 1 goes through a MA with $N = 200$. The plot shows that the MA curve elevates when people are present and approaches 0 when otherwise. However, it does not fully resemble human presence.

The PC evidences the closeness of the MA value to human presence data according to equation (16). We create five MA curves with different N values and observe which curve has the strongest PC coefficient. A matrix showing the PC of movement, five types of MAs, and human presence is shown in Figure 8. Payload is the feature name for movement data. The last row of the matrix shows the PC coefficient of presence with each feature. The highest value is 0.56, which is the MA at $N = 200$. Based on the interpretation, the curve has a moderate positive correlation with human presence. In comparison, the payload correlation is 0.36, which classifies as a low positive correlation.

A line plot can illustrate the growing trend of the PC coefficient based on the number of N values. The line plot is shown in Figure 9. The green line is the growth of the PC based on the increasing N values of the MA, where the red line is the PC of raw movement data. The MA method can increase the PC coefficient of movement features. However, using a data point too large will decrease the correlation. The optimum value is 200.

4.3. Training Classification Models. Because the MA of movement data has a moderate positive correlation with human presence, training a machine learning method with the new feature can hypothetically result in a model with good performance. In the model to be trained, the proposed input features are motion sensor data and some MA curves with different N values. The output class is human presence, with labels 1 for a human being present and 0 for no human present. It means that the type of classification is binary classification. We carry out an exhaustive test to find the optimum classification model for human presence based on movement data. The classification methods used are KNN, SVM, DT, and NB. EV is also applied to improve the performance of some of the mentioned methods.

The training process uses 50% of the dataset, while the testing stage uses the rest. It means there are 23436 training data and 23436 testing data. The dataset is shuffled prior to the data split to prevent uneven distribution. The test metrics are accuracy, precision, recall, and $F1$-score. Six initial features are used including five MA curves with a variation of N values: movement, MA ($N = 200$), MA ($N = 400$), MA ($N = 600$), MA ($N = 800$), and MA ($N = 1000$). The label is human presence. Cross-validation is also applied to

(1) Divide data into K equal folds
(2) **for** k in range $(0, K)$ **do**
(3) $R \leftarrow \text{Fold}_k$ in data
(4) $T \leftarrow$ data/R
(5) Train T
(6) $Acc_k \leftarrow$ evaluate R with trained model
(7) **end for**
(8) $Acc \leftarrow 1/K \sum_{k=1}^{K} Acc_k$

Algorithm 1: K-fold cross-validation.

Figure 5: Parts of the results of implementing the IoT architecture for smart lighting control based on human presence.

Table 1: Movement data collection specification.

Attribute	Value
Sampling rate	10 s
Collecting period	7 days
Collected data	56852 records
Feature	Movement
Feature values	1 (movement detected) 0 (no movement)
Label	Presence
Label values	1 (present) 0 (not present)

test the robustness of each model. A summary of the training specifications is given in Table 2.

In KNN, k describes the number of neighbors involved in calculating the closest distance between test and training data. A test of varying k finds the optimum KNN model. Changes in the value of accuracy, precision, recall, and $F1$-Score to the increase of k in KNN training is shown in Figure 10. The graph shows the comparison of the performance of the KNN model with $k = 1$ to $k = 5$. The values of precision and recall fluctuate, while the $F1$-score and accuracy values have a decreasing trend. Based on these tests, we conclude that $k = 1$ is the exact value for the optimum KNN model.

In SVM, the right type of kernel provides the optimum model. The kernel types compared are the linear kernel, 2^{nd}-degree polynomial, 3^{rd}-degree polynomial, RBF, and sigmoid. A comparison of the performance of the SVM classification model with five kernels is shown in Figure 11. The bar chart compares four performance values: accuracy, precision, recall, and $F1$-score. In all four metrics, sigmoid has the lowest performance. The 2^{nd}-degree polynomial has the highest recall but not the highest $F1$-score. The highest $F1$-score and accuracy go to the 3^{rd}-degree polynomial and RBF. However, the precision value of the 3^{rd}-degree polynomial is lower than RBF. Hence, the RBF kernel provides the optimum SVM model.

Using the right model depends on how to understand the data [60]. In 3.4, it has been explained that the selection of the SVM kernel depends on whether the data is linearly separable or not. In addition, the amount of data and the type of data also affect the selection of the model. A scatter plot matrix helps to understand the data better. The scatter plot matrix is often a tool for understanding high-dimensional data [61]. A visualization of the dataset in the form of a scatter plot matrix is shown in Figure 12. The scatter plot matrix shows that the scatter plot between each feature is not linearly separable. It explains why the linear kernel does not produce an optimum SVM model. Moreover, if the data is non-linearly separable and the RBF kernel is more optimum

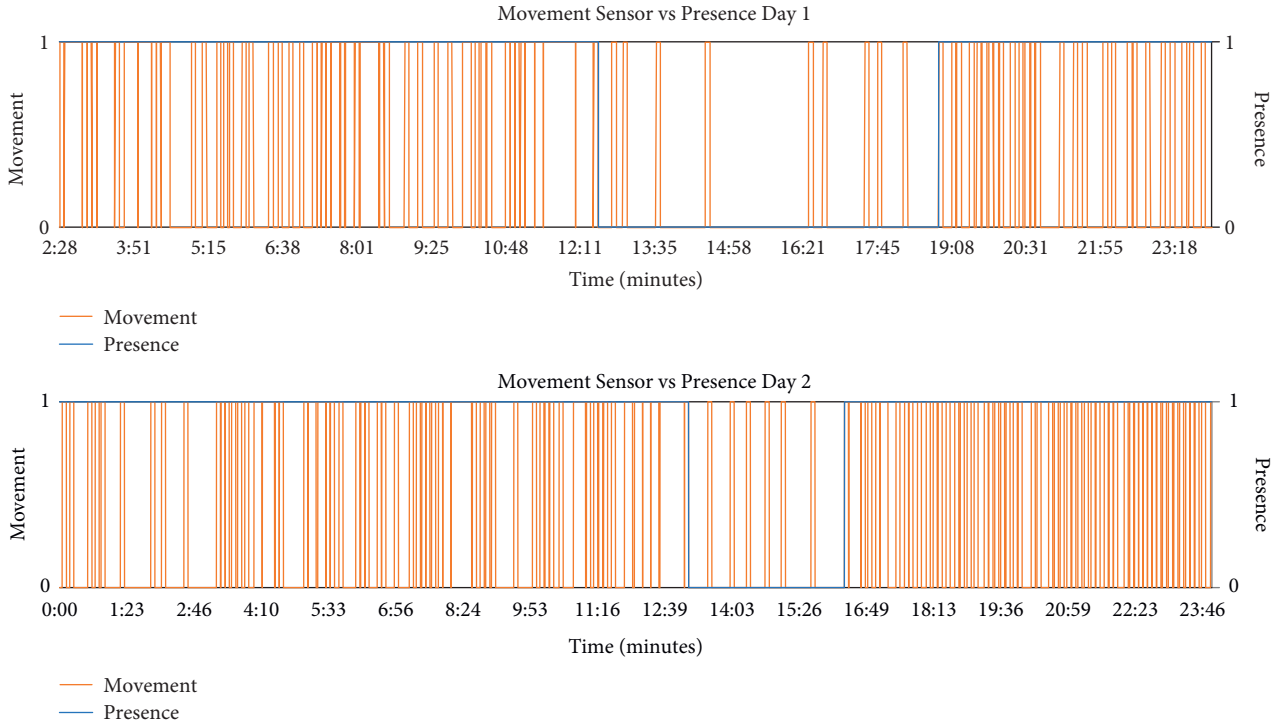

Figure 6: Partial snippet of the movement dataset with attendance labels.

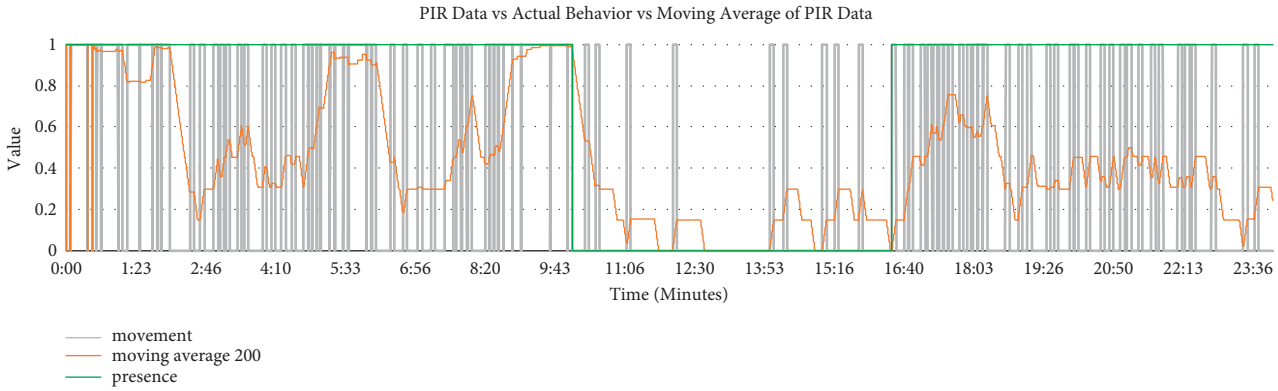

Figure 7: Visualization of the results of applying MA to the movement data and compared with actual human presence.

than other kernels, then the data is radially separable. In two dimensions, the binary class forms a doughnut shape, with one of the classes in the doughnut hole [62].

A too high max depth value in training can result in an overfitting DT model. The symptom of overfitting is that when comparing the performance of the tree with train data and cross-validation, the performance of the train data will continue to increase. In contrast, the cross-validation value will decrease or stagnate. Pruning is a solution to prevent overfitting the model. When using the early stop method in pruning, adding depths to the tree is stopped when the cross-validation value starts to drops [63]. The effect of increasing DT max depth on the accuracy of training data and validation data is shown in Figure 13. The orange line in the graph is the model's accuracy based on the train data, while the blue line is the average accuracy based on cross-validation. After the value of max depth = 12, the accuracy value of the cross-validation value decreases, so max depth = 12 is considered to provide the optimum DT model.

NB is a machine learning method that is more suitable for text-based analysis than classification on sensor data [64, 65]. It is also seen in this case when comparing the confusion matrix of KNN, SVM, DT, and NB. The confusion matrix of the four classifiers is shown in Figure 14. In the comparison, NB has the lowest performance. However, the FN and FP values of SVM, DTs, and NB are worth observing. The FP value of SVM is higher than its FN. However, it is the other way around in NB. They are peculiarities that EV can exploit, so we build it based on the four previous models. The test results complete the confusion matrix comparison. The

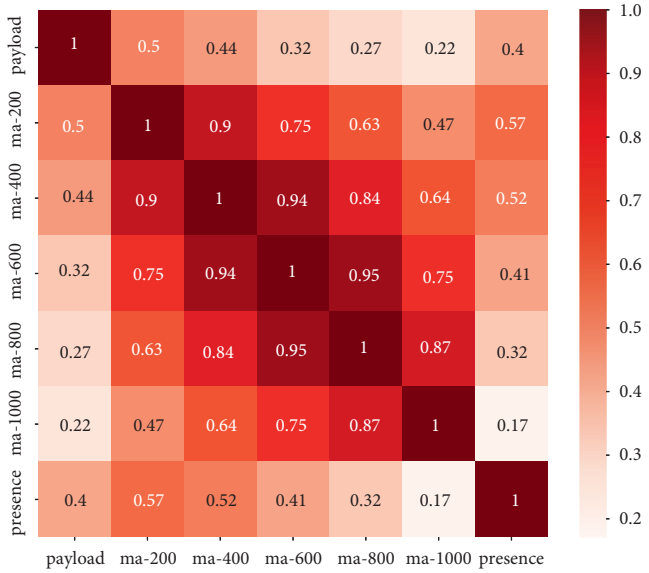

Figure 8: A matrix that displays the PC of movement (payload), five types of MAs, and presence.

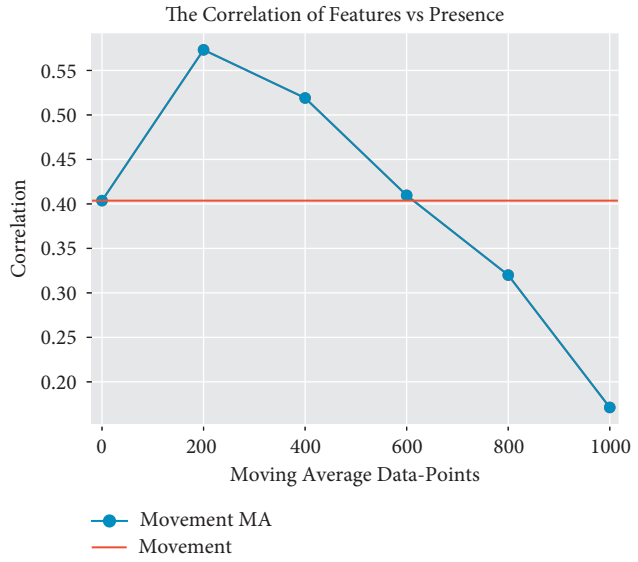

Figure 9: The growth and decline of the MA correlation with the increase in N values.

Table 2: Training specifications.

Attribute	Value
Machine learning methods	KNN, SVM, DT, NB, and EV
Split ratio	50 : 50
Training data	23436 records
Testing data	23436 records
Features	Movement, MA ($N = 200$), MA ($N = 400$), MA ($N = 600$), MA ($N = 800$), and MA ($N = 1000$)
Label	Presence
Data shuffle	On
Performance metrics	Accuracy, precision, recall, $F1$-score, and cross-validation

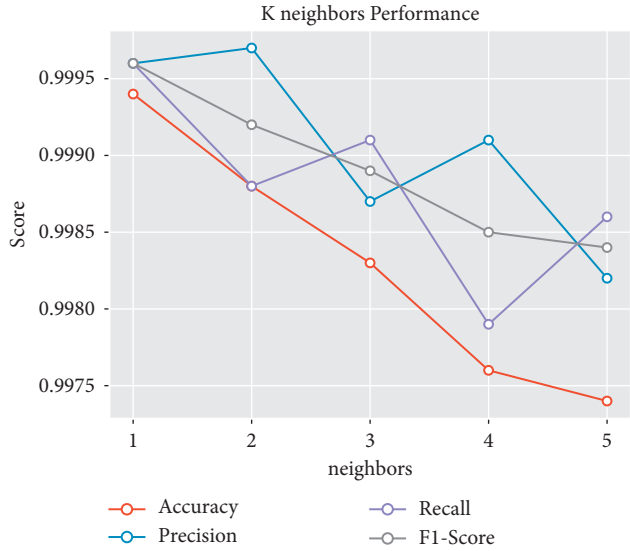

Figure 10: The increase of neighbors on the KNN performance.

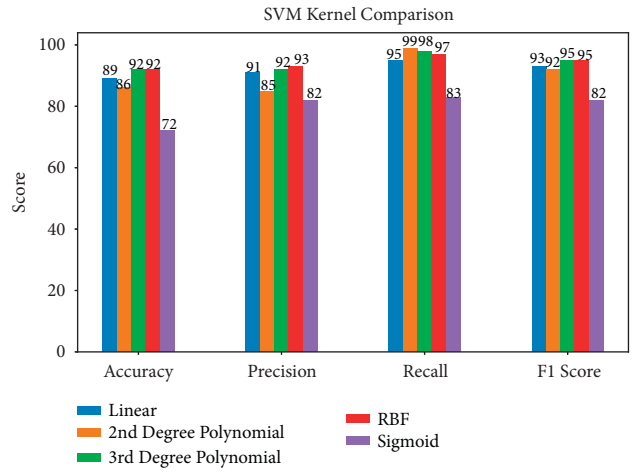

Figure 11: The performance comparison of different kernels on SVM classification.

results show that EV has a better confusion matrix than SVM and NB. In addition, the EV model has the lowest FP compared to SVM, DT, and NB. As a result, EV is a model that optimizes the NB model.

4.4. Performance Evaluation and Cross-Validation. Four optimum classification models can be compared: KNN with $k = 1$, SVM with RBF kernel, DT with max depth = 12, and EV from KNN, SVM, DT, and NB. The performance comparison of the four classifiers is shown in Figure 15. The comparison is in the form of a bar chart. In the bar chart, KNN is the blue bar, SVM is the orange bar, DT is the green bar, and EV is the red bar. Four metrics test the models: accuracy, precision, recall, and *F*1-score. SVM has the lowest performance in all four metrics of the four models. Between the three remaining models, EV is the only model with a recall value below 0.99. KNN excels in all four metrics, even compared to the DT. The optimum classification model for human presence based on movement data is KNN with $k = 1$.

The robustness of each model is also measured. We set SVM aside and only compare the models with the top three best performances from the previous tests, namely, KNN, DT, and EV. *K*-fold cross-validation measures the robustness of each model. The *K* values for testing are 2, 5, and 10 as they are commonly used [58]. Accuracy metric measures each cross-validation iteration. The cross-validation process accuracy comparison of the three models is shown in Figure 16. A box plot visualizes the performance comparison. In addition to the accuracy average, the box plot can also compare the accuracy variance of each case. For each model, the average accuracy trend increases as the number of folds increases. However, for EV specifically, the variance also increases. The EV owns the lowest average accuracy for each *K* value. At $K = 2$, the DT has the highest accuracy variance. For all *K* values, KNN has the lowest variance and the highest average. It concludes that KNN is also the most robust model apart from having the best performance.

The KNN model with $k = 1$ can still be optimized. Not all features will be related to the output class in machine learning. If an irrelevant feature enters the training process, what happens is garbage in, garbage out, and the performance of the model will drop [66]. Hence, at this stage, feature selection is carried out based on the PC value, previously calculated in 4.2. Assuming that increasing the number of uncorrelated features will reduce the performance of the classification model, the following scenarios are made based on the PC value and compared:

(i) 1 feature: MA ($N = 200$).

(ii) 2 features: MA ($N = 200$) and MA ($N = 400$).

(iii) 3 features: MA ($N = 200$), MA ($N = 400$), and MA ($N = 600$).

(iv) 4 features: MA ($N = 200$), MA ($N = 400$), MA ($N = 600$), and MA ($N = 800$).

(v) 5 features: MA ($N = 200$), MA ($N = 400$), MA ($N = 600$), MA ($N = 800$), and MA ($N = 1000$).

(vi) All features: all features included.

The effect of the number of features on the prediction performance of the KNN model is shown in Figure 17. The image is in the form of a line plot. The four metrics compared include accuracy, precision, recall, and *F*1-score. Results show an increasing trend in the addition of the number of features. It proves that although the MA with $N > 200$ has a lower correlation than the MA with $N = 200$, these features are still relevant in classifying human presence. Subsequently, the model with five features and six features has the same performance. The conclusion is that if the raw movement data departs the dataset, the model performance does not decrease, reducing complexity. Hence, the feature selection process result concludes that the KNN model with five features is the optimum KNN model.

For example, two features have a high correlation with the output class. In the understanding of multicollinearity, if

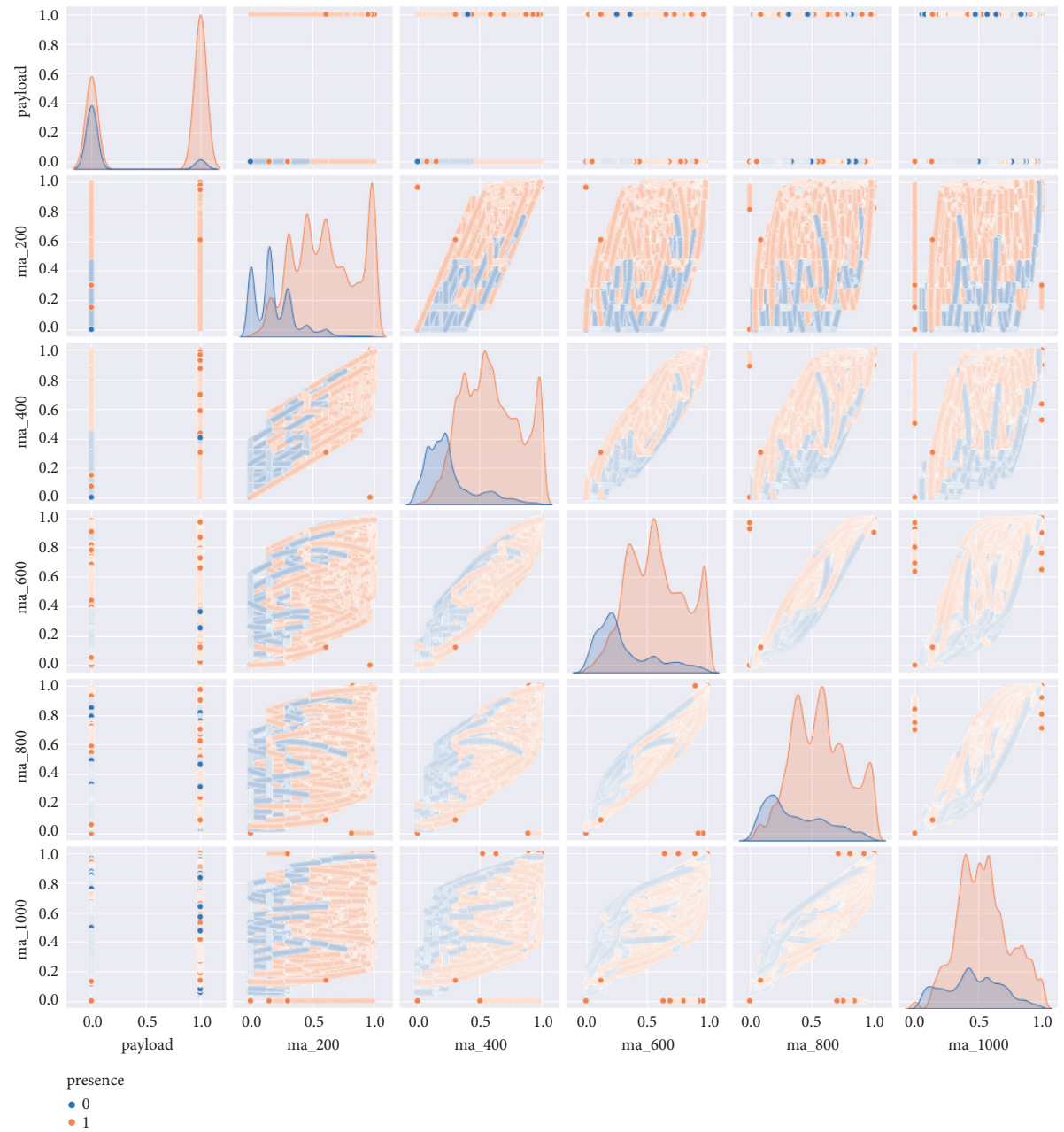

Figure 12: The scatter plot matrix of all features.

the two features have a high correlation and one of them is not excluded, the performance of the model will become poor, especially the linear regression model [67]. For example, revisiting the PC matrix in Figure 8, MA ($N = 600$) is highly correlated to MA ($N = 800$). We investigate this by applying the moving PC to the time-series dataset. A visualization of the application of the moving PC with $N = 6000$ to the dataset can be seen in Figure 18. We take a snapshot of two different cases. The upper part of the image is a situation where there is not much fluctuation in attendance. In this situation, MAs with high N have a high correlation. The bottom part of the image is a situation where there is much fluctuation in human presence. The MA with low N has a high correlation in this situation. It explains why the 5-feature model has the best performance.

We use the test data to measure how directly using PIR sensor movement data to lighting control would perform. We call it the raw method. The significance of the presence classification model (proposed method) on the raw method appears in a side-to-side comparison. The comparison of the two methods is shown in Figure 19. The image is in the form of a bar plot. It shows accuracy, precision, recall, and $F1$-score, where the proposed method is the blue bar, and raw is the orange bar. The two biggest significances are accuracy and recall, 99.7% to 67.8% and 99.8% to 62.6%, respectively.

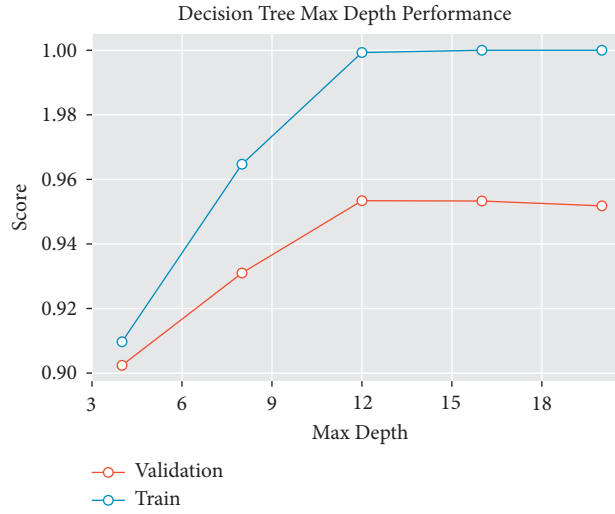

Figure 13: The increase of max depth on the DT accuracy.

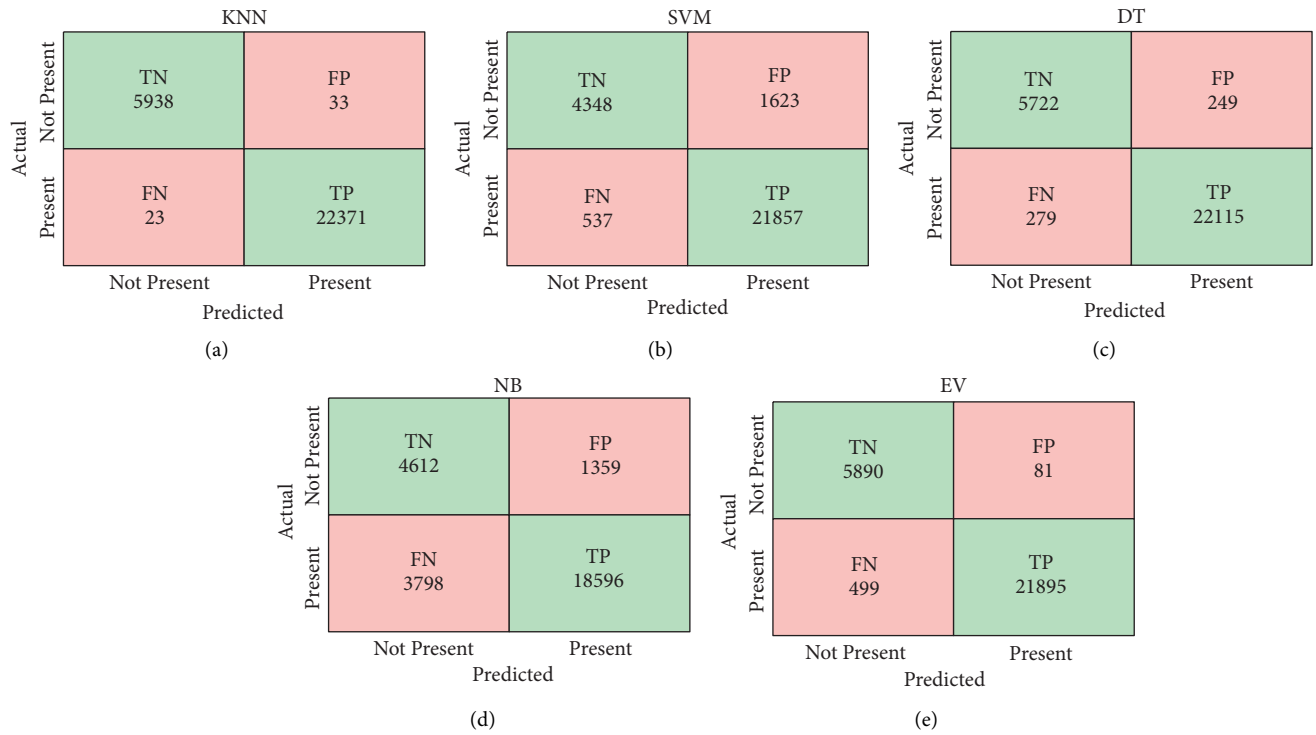

Figure 14: The confusion matrix of all classification models: (a) KNN, (b) SVM, (c) DT, (d) NB, and (e) EV.

Visualization can showcase the performance of the KNN model in predicting human presence. The visualization compares the actual time-series attendance and the predicted time-series attendance. The comparison of the two and also movement data with sensors is shown in Figure 20. The top part of the image is the time-series presence of the PIR sensor measurement results. Then, the middle part of the image is the actual attendance time series. The last part of the image below is the time series of the prediction results of the KNN model. When compared between the movement data from the PIR sensor and the presence data from the KNN model predictions, the latter is more in line with the actual presence data.

5. Discussion

In the test results, the application of MAs to the movement data of the motion sensor results can increase the PC of the features on the actual presence of humans in the room. This is in accordance with existing studies, namely, [20–23] and [24]. The related studies use MA to increase the correlation

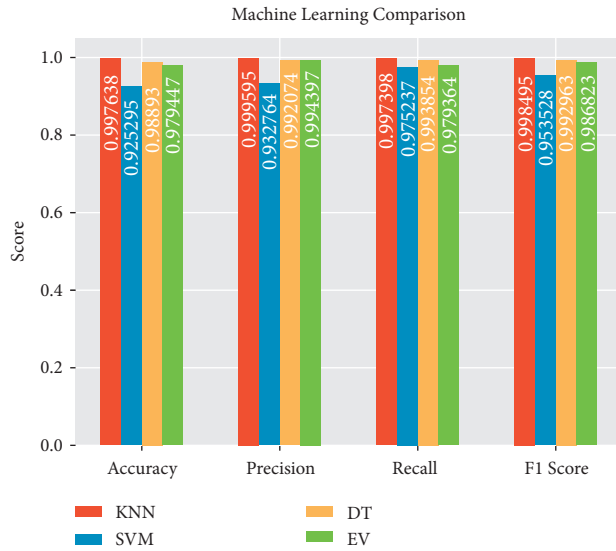

Figure 15: A performance comparison of four classification models in predicting human presence.

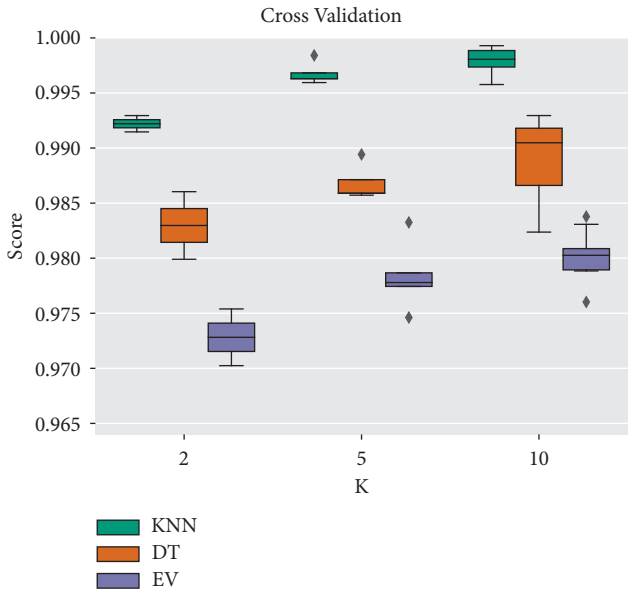

Figure 16: The cross-validation accuracy results on K-folds with $K = 2$, 5, and 10.

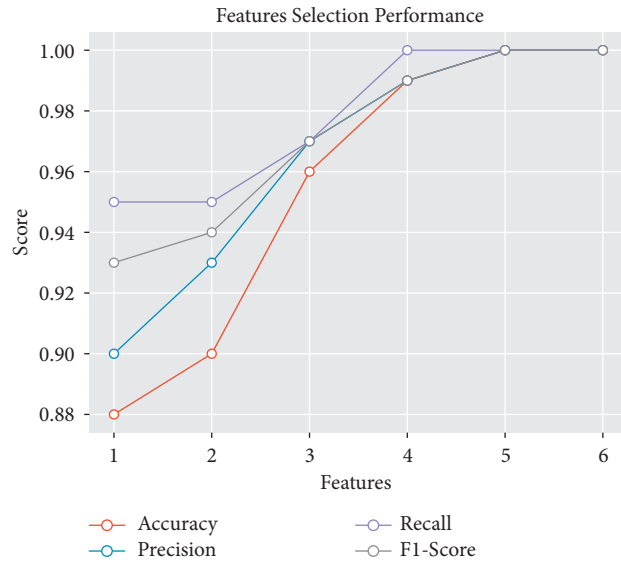

Figure 17: The increase of features on the classification model performance.

of regression and classification features, among others, for noise reduction and forecasting.

The reason why KNN can be better than SVM, DT, NB, and EV is the nature of KNN, which is robust against noisy data [68]. SVM with RBF kernel is indeed good for radially separable data. However, if the data has high variance, then it possibly affects the performance of both SVM and DT in performing data separation.

We make direct comparisons of our proposed method with related studies to emphasize the contribution and novelty of our proposed method. The related studies are human presence-based smart lighting control using other equipment. The comparison is given in Table 3. The superior values of each column are made bold. Compared to the benchmarked studies, our proposed method has the best performance, 99.8%. The research [11] has an approximate result, which is 99.3%. The study uses a concept similar to a MA, namely, a sliding window to calculate several statistical features such as mean, standard deviation, and max. A random forest RF model is an optimum model that applies the sliding window feature in the study. However, it uses several different sensors, some of which are expensive sensors, such as smartwatches and real-time location systems. Studies that also use expensive devices for activity recognition are [27], which uses a depth camera; [18], which uses five monochrome cameras; and [12], which uses Switchmate.

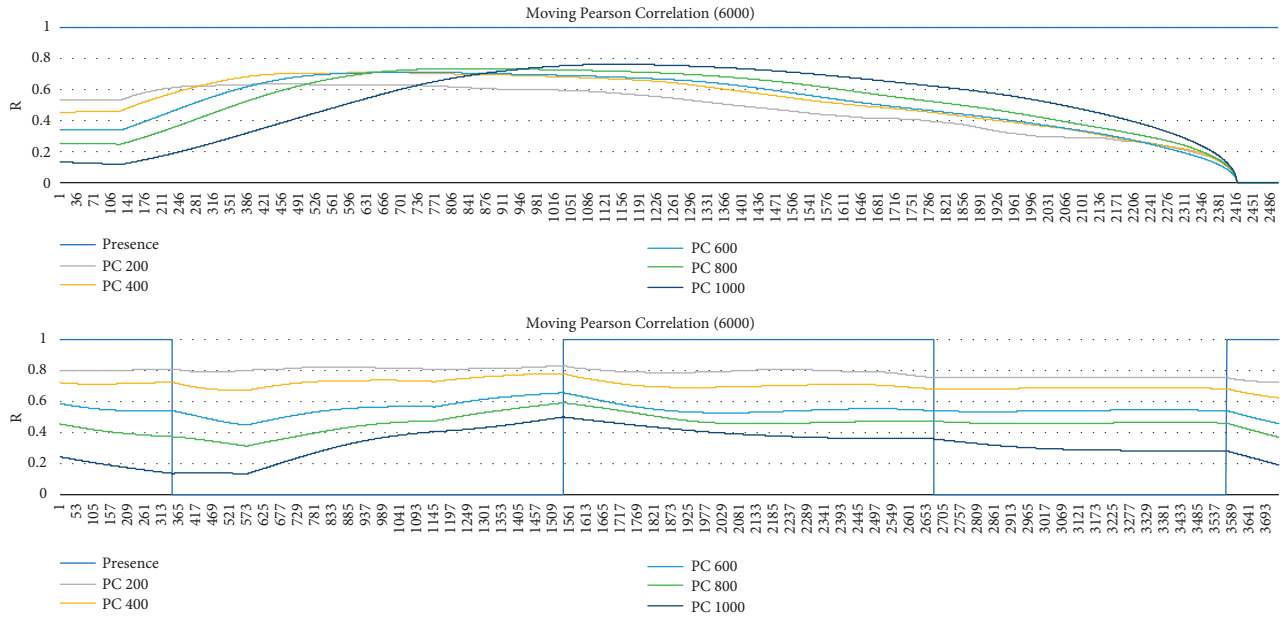

Figure 18: Visualization of the application of the moving PC ($N = 6000$) to the dataset.

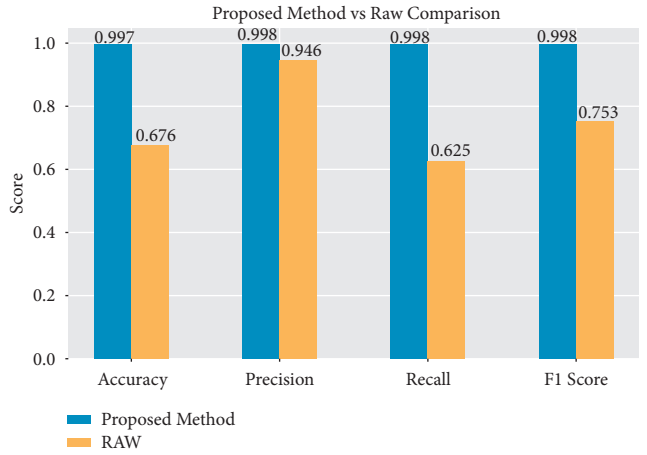

Figure 19: A performance comparison of classification with the KNN model (proposed method) with decision-making directly done on motion sensor (raw method).

The latter research does not use accuracy in calculating performance but uses LUR. LUR is their proposed metric that describes the ratio between the time the lights are on and the time someone is present in the room. We assume that LUR is equivalent to accuracy. Our proposed method is the method with the best performance and a low-cost solution.

Moreover, we also investigate the factors that influence performance in studies regarding PIR sensors in human presence-based smart lighting control. The comparison of these related works is shown in Table 4. Based on our proposed method and [9], it seems that there is a negative relationship between the number of activities and performance. However, [5] that has 14 activities has a better performance than [10, 17] that only have five activities. In addition, our proposed method and [9] are location-based methods. A person's presence is determined based on whether the person is under sensor or not. Meanwhile, [5, 10] and [17] that have significantly lower performance are not location-based. These methods define an activity where the activity is independent of its location. Further research can investigate the performance of a PIR sensor-based activity recognition on determining activities that are location-based.

In future work, the direction of this research is to increase user comfort from smart lighting. Hence, if the automatic light control is carried out based on the presence of people, people will not feel discomfort. For the long term, the research aims to measure user comfort when users use smart lighting that has applied the novel method in this research. The user comfort method proposal is a novel one, which is a quantitative method.

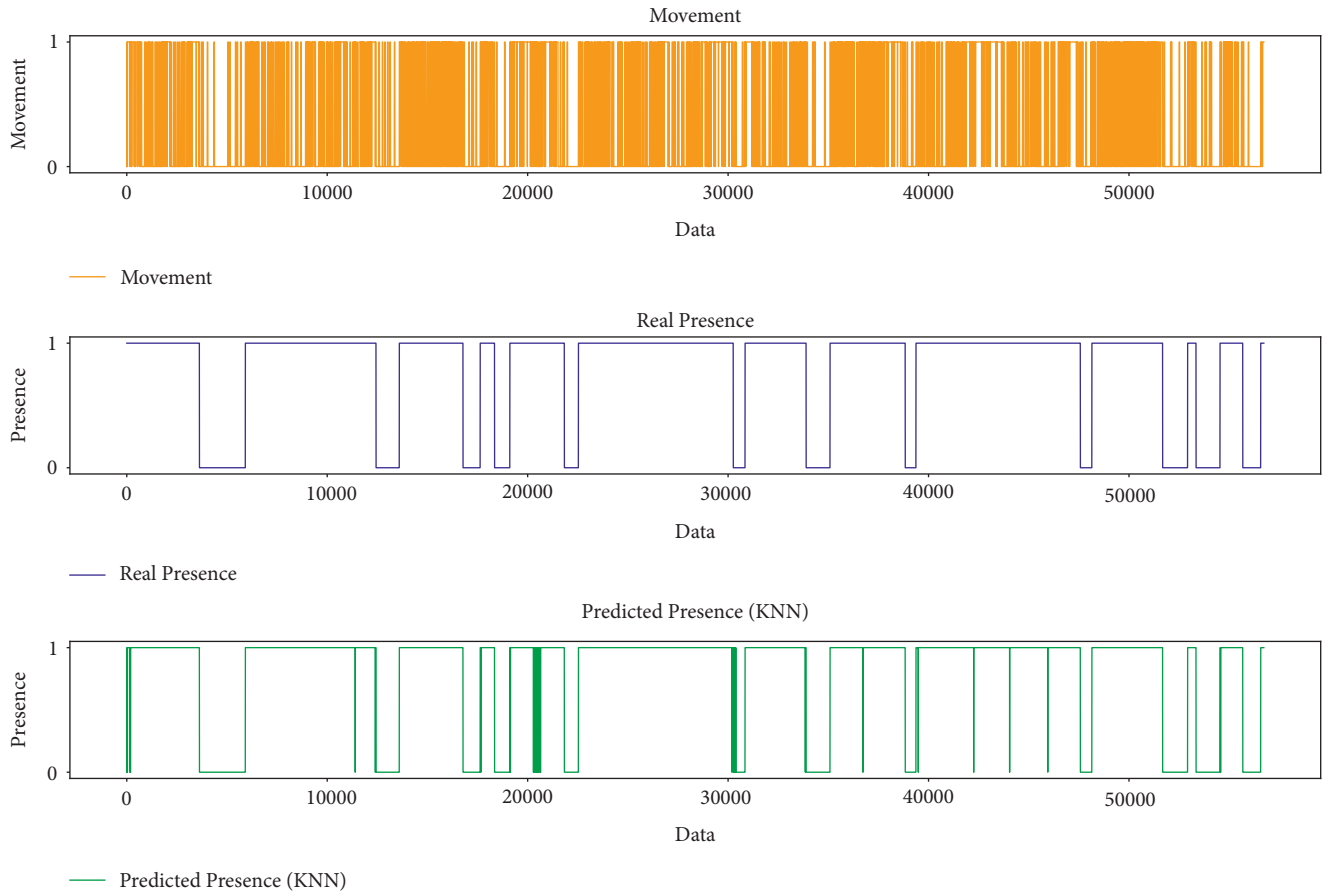

Figure 20: A time-series comparison of the movement data from the PIR sensor, the real presence, and the predicted presence by the KNN classification model.

Table 3: A Comparison of related works on human presence-based smart lighting control.

Reference	Equipment	Cost	Accuracy (%)
Proposed method	PIR sensor	**US$13**	**99.8**
Lupion et al. [11]	PIR sensor, pressure sensor, switch sensor, smartwatches, RTLS	US$65	99.3
Park et al. [12]	Light sensor, switchmate	US$68	67
Dai et al. [18]	Five monochrome cameras	US$300[1]	90.2
Chun et al. [19]	Depth camera	US$36[1]	78.3

[1]Estimated.

Table 4: A comparison of related studies using the PIR sensor for smart lighting control.

Reference	Method	Number of activities	Location-based	Accuracy (%)
Proposed method	CIMA	2	Yes	**99.8**
Ramadhan et al. [5]	HHMM	14	No	93
Jin et al. [9]	TS-ANN	2	Yes	97
Fakhruddin et al. [10]	PCA-KNN	5	No	94
Putrada et al. [17]	HHMM	5	No	87.6

Furthermore, the next future work is to use this novel method to monitor the movement of people in the house. This achievement leads to a novel predictive control of lights based on the user movement. The benefit of this proposal is that automatic light control can occur without the user being aware of it. As an illustration of the case, before people enter the room, the lights are already on. It will furtherly increase user comfort while still maintaining energy efficiency.

6. Conclusions

This paper proposes CIMA, a novel classification-integrated moving average model for smart lighting intelligent control based on human presence. A smart lighting system based on the Internet of things (IoT) applies the proposed method. It uses passive infrared (PIR) sensors, light-emitting diode (LED) lights, relays, NodeMCU, Raspberry Pi, and supporting software. In the PC test, the movement data from the PIR sensor has a correlation of 0.36 to attendance, while the moving average (MA) correlation to human presence can reach 0.56. In exhaustive testing of machine learning classification methods, k-nearest neighbor (KNN) is the model with the best and most robust performance with an accuracy value of 99.8%. It is more accurate than direct light control decisions based on motion sensors with 67.6%. We conclude that our proposed method can increase the correlation value of movement features on attendance. At the same time, an accurate and robust KNN classification model is applicable for human presence-based smart lighting intelligent control.

Data Availability

Data supporting reported results can be found at https://doi.org/10.34820/FK2/8BXAYW.

Conflicts of Interest

The authors declare no conflicts of interest.

Authors' Contributions

All authors contributed equally to this manuscript.

Acknowledgments

The authors would like to thank Telkom University and the Ministry of Education, Culture, Research, and Technology for fully funding this research through the Doctoral Dissertation Research scheme and other supporting funds.

References

[1] M. Soheilian, G. Fischl, and M. Aries, "Smart lighting application for energy saving and user well-being in the residential environment," *Sustainability*, vol. 13, no. 11, p. 6198, 2021.

[2] P. Smallwood, *Lighting, leds and smart lighting market overview*, US Dept. Energy SSL Workshop, Raleigh, NC, USA, 2016.

[3] M. Füchtenhans, E. H. Grosse, and C. H. Glock, "Smart lighting systems: state-of-the-art and potential applications in warehouse order picking," *International Journal of Production Research*, vol. 59, no. 12, pp. 3817–3839, 2021.

[4] O. O. Ordaz-García, M. Ortiz-Lopez, F. J. Quiles-Latorre, J. G. Arceo-Olague, R. Solis-Robles, and F. J. Bellido-Outeirino, "DALI bridge FPGA-based implementation in a wireless sensor Node for IoT street lighting applications," *Electronics*, vol. 9, no. 11, p. 1803, 2020, https://www.mdpi.com/2079-9292/9/11/1803.

[5] R. Nur Ghaniaviyanto, "Aji Gautama Putrada, and Maman Abdurohman. Improving smart lighting with activity recognition using hierarchical hidden Markov model," *Indonesia Journal on Computing (Indo-JC)*, vol. 4, no. 2, pp. 43–54, 2019.

[6] L. Gyu Myoung and J. Yun Kim, "The internet of things—a problem statement," in *Proceedings of the 2010 International Conference on Information and Communication Technology Convergence (ICTC)*, Jeju, Korea (South), November 2010.

[7] J. Guo, Y. Mu, M. Xiong, Y. Liu, and J. Gu, "Activity feature solving based on tf-idf for activity recognition in smart homes," *Complexity*, vol. 2019, pp. 1–10, 2019.

[8] M. José, E. Irigoyen, and V. M. Becerra, "Intelligent control approaches for modeling and control of complex systems," *Complexity*, vol. 2018, pp. 1-2, 2018.

[9] Y. Jin, D. Yan, X. Zhang, J. An, and M. Han, "A data-driven model predictive control for lighting system based on historical occupancy in an office building: methodology development," *In Building Simulation*, vol. 14, pp. 219–235, 2021.

[10] R. Irsyad Fakhruddin, "Maman Abdurohman, and Aji Gautama Putrada. Improving pir sensor network-based activity recognition with pca and knn," in *Proceedings of the 2021 International Conference On Intelligent Cybernetics Technology & Applications (ICICyTA), Pages 138–143* IEEE, Bandung, Indonesia, December 2021.

[11] M. Lupión, P. M. Ortigosa, Q. Javier Medina, J. Medina-Quero, and J. F. Sanjuan, "Dolars, a distributed on-line activity recognition system by means of heterogeneous sensors in real-life deployments—a case study in the smart lab of the university of almería," *Sensors*, vol. 21, no. 2, p. 405, 2021.

[12] J. Y. Park, T. Dougherty, H. Fritz, and Z. Nagy, "LightLearn: an adaptive and occupant centered controller for lighting based on reinforcement learning," *Building and Environment*, vol. 147, pp. 397–414, 2019.

[13] J. Lei, X. Wang, Y. Zhang, L. Zhu, and L. Zhang, "Policy and law assessment of covid-19 based on smooth transition autoregressive model," *Complexity*, vol. 2021, Article ID 6659117, 13 pages, 2021.

[14] L. Borzì, S. Fornara, F. Amato, G. Olmo, C. A. Artusi, and L. Lopiano, "Smartphone-based evaluation of postural stability in Parkinson's disease patients during quiet stance," *Electronics*, vol. 9, no. 6, p. 919, 2020, https://www.mdpi.com/2079-9292/9/6/919.

[15] H. Layth Rafea, "Four classification methods naïve bayesian, support vector machine, k-nearest neighbors and random forest are tested for credit card fraud detection," *Master's thesis, Altınbaş Üniversitesi*, 2018.

[16] K. Raza, "Improving the prediction accuracy of heart disease with ensemble learning and majority voting rule," in *U-healthcare Monitoring Systems,* vol. 1, pp. 179–196, Elsevier, 2019.

[17] P. Aji Gautama, "Nur Ghaniaviyanto Ramadhan, and MA Makky. An evaluation of activity recognition with hierarchical hidden Markov model and other methods for smart lighting in office buildings," *ICIC International*, vol. 16, 2022.

[18] Ji Dai, J. Wu, B. Saghafi, J. Konrad, and P. Ishwar, "Towards privacy-preserving activity recognition using extremely low temporal and spatial resolution cameras," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 68–76, Boston, 2015.

[19] S. Y. Chun, C.-S. Lee, and J.-S. Jang, "Real-time smart lighting control using human motion tracking from depth camera," *Journal of Real-Time Image Processing*, vol. 10, no. 4, pp. 805–820, 2015.

[20] A. Husnayain, A. Fuad, and L. Lazuardi, "Correlation between google trends on dengue fever and national surveillance

report in Indonesia," *Global Health Action*, vol. 12, no. 1, Article ID 1552652, 2019.

[21] Z. Hu, Y. Zhang, Y. Zhao et al., "A water quality prediction method based on the deep lstm network considering correlation in smart mariculture," *Sensors*, vol. 19, no. 6, p. 1420, 2019.

[22] Yu Peng, S. Long, J. Ma, J. Song, and Z. Liu, "Temporal-spatial variability in correlations of drought and flood during recent 500 years in inner Mongolia, China," *Science of the Total Environment*, vol. 633, pp. 484–491, 2018.

[23] H. S. Badr, E. Dong, M. M. Squire et al., "Association between mobility patterns and covid-19 transmission in the USA: a mathematical modelling study," *The Lancet Infectious Diseases*, vol. 20, no. 11, pp. 1247–1254, 2020.

[24] O. P. Singh, T. A. Howe, and M. B. Malarvili, "Real-time human respiration carbon dioxide measurement device for cardiorespiratory assessment," *Journal of Breath Research*, vol. 12, no. 2, 2018.

[25] P. Aji Gautama, "Maman Abdurohman, Doan Perdana, and Hilal Hudan Nuha. Machine learning methods in smart lighting towards achieving user comfort: a survey," *IEEE Access*, vol. 10, 2022.

[26] P. Aji Gautama and A. Maman, "Anomaly detection on an iot-based vaccine storage refrigerator temperature monitoring system," in *Proceedings of the 2021 International Conference On Intelligent Cybernetics Technology & Applications (ICICyTA)*, pp. 75–80, IEEE, Bandung, Indonesia, December 2021.

[27] S. Chun and C.-S. Lee, "Applications of human motion tracking: smart lighting control," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pp. 387–392, Portland, 2013.

[28] S. Fuada, T. Adiono, and L. Siregar, "Internet-of-things for smart street lighting system using esp8266 on mesh network,".

[29] T. Montanaro, I. Sergi, A. Motroni et al., "An iot-aware smart system exploiting the electromagnetic behavior of uhf-rfid tags to improve worker safety in outdoor environments," *Electronics*, vol. 11, no. 5, p. 717, 2022, https://www.mdpi.com/2079-9292/11/5/717.

[30] P. Kumar, P. Rai, and H. B. Yadav, "Smart lighting and switching using Internet of Things," in *Proceedings of the 2021 11th International Conference on Cloud Computing, Data Science & Engineering (Confluence), pages 536–539*, IEEE, Noida, India, January 2021, https://ieeexplore.ieee.org/document/9377078/.

[31] E. Juškevičius, "Smart home lighting system using iot technologies," Bachelor Thesis, South Eastern Finland University of Applied Science, Kouvola, 2021.

[32] J. Purmaissur, A. Seeam, S. Guness, and X. Bellekens, "Augmented reality intelligent lighting smart spaces," in *Proceedings of the 2019 Conference On Next Generation Computing Applications (NextComp)*, pp. 1–5, IEEE, Mauritius, September 2019.

[33] P. Aji Gautama and P. Doan, "Improving thermal Camera Performance in Fever Detection during Covid-19 Protocol with Random forest Classification," in *Proceedings of the 2021 International Conference Advancement in Data Science, E-learning and Information Systems (ICADEIS)*, pp. 1–6, IEEE, Bali, Indonesia, October 2021.

[34] G. S. Smrithy, R. Balakrishnan, and N. Sivakumar, "Anomaly detection using dynamic sliding window in wireless body area networks," *Data Science and Big Data Analytics*, Springer, pp. 99–108, Singapore.

[35] G. Kechyn, L. Yu, Y. Zang, and S. Kechyn, "Sales forecasting using wavenet within the framework of the kaggle competition," 2018, https://arxiv.org/abs/1803.04037.

[36] M. Vijh, D. Chandola, V. A. Tikkiwal, and A. Kumar, "Stock closing price prediction using machine learning techniques," *Procedia Computer Science*, vol. 167, pp. 599–606, 2020.

[37] T. Duong Truong, "Nguyen Ba Hoang Quan, and Nopadon Maneetien. Implementation of Moving Average Filter on Stm32f4 for Vibration Sensor Application," in *Proceedings of the 2018 4th International Conference on Green Technology and Sustainable Development (GTSD)*, pp. 627–631, IEEE, Ho Chi Minh City, Vietnam, November 2018.

[38] F. Ghassani, M. Abdurohman, and A. G. Putrada, "Prediction of Smarthphone Charging Using K-Nearest Neighbor Machine Learning," in *Proceedings of the 2018 Third International Conference on Informatics and Computing (ICIC)*, pp. 1–4, IEEE, Palembang, Indonesia, October 2018.

[39] V. K. Chauhan, K. Dahiya, and A. Sharma, "Problem formulations and solvers in linear SVM: a review," *Artificial Intelligence Review*, vol. 52, no. 2, pp. 803–855, 2019.

[40] G. A. Rattá, J. Vega, A. Murari, and J. Efda, "Improved feature selection based on genetic algorithms for real time disruption prediction on jet," *Fusion Engineering and Design*, vol. 87, no. 9, pp. 1670–1678, 2012.

[41] M. Gelfusa, A. Murari, M. Lungaroni et al., "A support vector machine approach to the automatic identification of fluorescence spectra emitted by biological agents. In Optics and Photonics for Counterterrorism," *Crime Fighting, and Defence XII*, vol. 9995, 2016.

[42] S. Ghosh, A. Dasgupta, and A. Swetapadma, "A study on support vector machine based linear and non-linear pattern classification," in *Proceedings of the 2019 International Conference On Intelligent Sustainable Systems (ICISS)*, pp. 24–28, IEEE, Palladam, India, February 2019.

[43] A. Taufiqurrahman, A. Gautama Putrada, and F. Dawani, "Decision tree regression with adaboost ensemble learning for water temperature forecasting in aquaponic ecosystem," in *Proceedings of the 2020 6th International Conference On Interactive Digital Media (ICIDM)*, pp. 1–5, IEEE, Bandung, Indonesia, December 2020.

[44] A. N. Iman, A. G. Putrada, S. Prabowo, and D. Perdana, "Peningkatan ktmrpp," *Jurnal Elektro dan Telekomunikasi Terapan*, vol. 8, no. 1, pp. 978–985, 2021.

[45] K. Yusuf, M. Abdurohman, and A. G. Putrada, "Increasing Passive Rfid-Based Smart Shopping Cart Performance Using Decision Tree," in *Proceedings of the 2019 5th International Conference on Computing Engineering and Design (ICCED)*, pp. 1–5, IEEE, Singapore, April 2019.

[46] H. Bagaskara, A. Gautama Putrada, and E. Ariyanto, "Proximity and dynamic device pairing based authentication for iot end devices with decision tree method," in *Proceedings of the 2020 6th International Conference On Interactive Digital Media (ICIDM)*, pp. 1–5, IEEE, Bandung, Indonesia, December 2020.

[47] T. Daniya, M. Geetha, and K. Suresh Kumar, "Classification and regression trees with gini index," *Advances in Mathematics: Scientific Journal*, vol. 9, no. 10, pp. 8237–8247, 2020.

[48] D. Berrar, "Bayes' theorem and naive bayes classifier. Encyclopedia of Bioinformatics and Computational Biology: ABC of Bioinformatics," *Elsevier Science Publisher: amsterdam, The Netherlands*, vol. 403, 2018.

[49] O. Sagi and L. Rokach, "Ensemble learning: a survey," *WIREs Data Mining and Knowledge Discovery*, vol. 8, no. 4, Article ID e1249, 2018.

[50] F. T. Breiner, M. P. Nobis, A. Bergamini, and A. Guisan, "Optimizing ensembles of small models for predicting the distribution of species with few occurrences," *Methods in Ecology and Evolution*, vol. 9, no. 4, pp. 802–808, 2018.

[51] S. Mani, S. Kumari, A. Jain, and P. Kumar, "Spam review detection using ensemble machine learning," in *Proceedings of the International Conference on Machine Learning and Data Mining in Pattern Recognition*, pp. 198–209, Springer, Cham, July 2018.

[52] B. H. Al-Zadid Sultan and T. Tanpia, "An ensemble hard voting model for cardiovascular disease prediction," in *Proceedings of the 2020 2nd International Conference On Sustainable Technologies For Industry 4.0 (STI)*IEEE, Dhaka, Bangladesh, December 2020.

[53] O. Esezi Isaac and C. A. Eric, "Test for significance of pearson's correlation coefficient," *International Journal of Innovative Mathematics, Statistics & Energy Policies*, vol. 6, no. 1, pp. 11–23, 2018.

[54] S. Muhammad Bagus, "Aji Gautama Putrada, and Maman Abdurohman. Evaluation of face detection and recognition methods in smart mirror implementation," in *Proceedings of Sixth International Congress on Information and Communication Technology*, pp. 449–457, Springer, Singapore, September 2022.

[55] Y. Demir, N. Ö. Atar, Ü. Güzelküçük, K. Aydemir, and E. Yaşar, "The use of and satisfaction with prosthesis and quality of life in patients with combat related lower limb amputation, experience of a tertiary referral amputee clinic in Turkey," *Age*, vol. 61, no. 1, pp. 6–10, 2019.

[56] Y. Liu, Y. Mu, K. Chen, Y. Li, and J. Guo, "Daily activity feature selection in smart homes based on pearson correlation coefficient," *Neural Processing Letters*, vol. 51, no. 2, pp. 1771–1787, 2020.

[57] M. Belkin, D. J. Hsu, and P. Mitra, "Overfitting or perfect fitting? risk bounds for classification and regression rules that interpolate," *Advances in Neural Information Processing Systems*, vol. 31, 2018.

[58] B. Ghojogh and M. Crowley, "The theory behind overfitting, cross validation, regularization, bagging, and boosting: tutorial," 2019, https://arxiv.org/abs/1905.12787.

[59] F. Khan, A. Urooj, K. Ullah, B. Alnssyan, and Z. Almaspoor, "A comparison of autometrics and penalization techniques under various error distributions: evidence from Monte Carlo simulation," *Complexity*, vol. 2021, Article ID 9223763, 8 pages, 2021.

[60] Y. Gil, J. Honaker, S. Gupta et al., "Towards human-guided machine learning," in *Proceedings of the 24th international conference on intelligent user interfaces*, pp. 614–624, California, Marina del Ray, March 2019.

[61] K. Asai, T. Fukusato, and T. Igarashi, "An interactive tool for feature analysis of outliers in multi-dimensional data," Research Gate, 2018.

[62] G. Lovell, "Unified model combining the boundary conditions encountered at the input, the carrier optical fiber and the output in spatially multiplexed optical communication channels," *PhD thesis, Florida institute of Technology*, Melbourne, 2019.

[63] T.-Yi Chen, Y.-H. Chang, M.-C. Yang, and H.-W. Chen, "How to cultivate a green decision tree without loss of accuracy," in *Proceedings of the ACM/IEEE international symposium on low power electronics and design*, pp. 1–6, Massachusetts, Boston, August 2020.

[64] S. He, Y. He, and M. Li, "Classification of illegal activities on the dark web," in *Proceedings of the 2019 2nd international conference on information science and systems*, pp. 73–78, Tokyo, Japan, March 2019.

[65] O. Cherqi, G. Mezzour, M. Ghogho, and M. E. Koutbi, "Analysis of Hacking Related Trade in the Darkweb," in *Proceedings of the 2018 IEEE international conference on intelligence and security informatics (ISI)*, pp. 79–84, IEEE, Miami, FL, USA, November 2018.

[66] E. Victor, J. Staartjes, M. Kernbach et al., "Foundations of feature selection in clinical prediction modeling," in *Proceedings of the Machine Learning in Clinical Neuroscience*, pp. 51–57, Springer, Cham, 2022.

[67] B. Shi, B. Meng, H. Yang, J. Wang, and W. Shi, "A novel approach for reducing attributes and its application to small enterprise financing ability evaluation," *Complexity*, vol. 2018, Article ID 1032643, 17 pages, 2018.

[68] S. Alfin Pratama, "Kusuma Ayu Laksitowening, and Ibnu Asror. Time series prediction on college graduation using knn algorithm," in *Proceedings of the 2020 8th International Conference on Information and Communication Technology (ICoICT)*, pp. 1–4, IEEE, Yogyakarta, Indonesia, June 2020.