



Smart Hostel Management System with Facial Recognition and Predictive Behavioural Analytics

Goh Qin Nee^{ID*}¹, and Justtina Anantha Jothi John^{ID}¹

¹ Department of Computing, UOW Malaysia KDU Penang College, George Town, Malaysia.

KEYWORDS

Hostel Management System
Facial Recognition
Predictive Analytics
DeepFace
Decision Tree

ARTICLE HISTORY

Received 10 Jan 2026
Received in revised form
7 Jul 2026
Accepted 31 Aug 2026
Available online 23 Sep 2026

ABSTRACT

The higher education sector in Malaysia is expanding rapidly, aiming to attract 250,000 international students by 2025. With this rapid growth, university hostels require a more efficient management system to ensure the safety and well-being of residents. However, many hostels still rely on paper-based logs or basic RFID systems. These traditional methods are prone to human error and are at risk of proxy attendance. Furthermore, these systems are often passive and fail to analyse resident data to identify behavioural risks. The aim of this project was to investigate these limitations and develop a Smart Hostel Management System that integrates biometric authentication using the DeepFace framework and predictive analytics. The specific objective was to build a centralized platform that automates entry/exit logging to eliminate identity spoofing and utilize the Decision Tree machine learning model to predict resident risk levels based on behavioural history. The research proved that integrating Convolutional Neural Networks (CNN) with decision-based learning can effectively replace manual monitoring.

© 2026 The Authors. Published by Penteract Technology.

This is an open access article under the CC BY-NC 4.0 license (<https://creativecommons.org/licenses/by-nc/4.0/>).

1. INTRODUCTION

For many students, university life represents a step toward independence. This makes hostels a vital support system in fostering community and personal growth. As Malaysia positions itself as a global education hub, aiming to attract 250,000 international students by 2025[1], the demand for high quality student accommodation is rising. However, while institutions expand physical infrastructure, administrative systems often remain outdated.

Many universities in Malaysia are still relying on manual paper based processes for most of the processes like entry/exit logging, room allocation and disciplinary monitoring. While some institutions adopted digital solutions like RFID cards, these systems still suffer from their own vulnerability. Furthermore, traditional systems are passive, meaning they record historical data but still lack the intelligence to analyse patterns or predict potential behaviour risks among residents.

The purpose of this paper is to discuss the role of automation in modernizing hostel administration and the framework for implementing intelligent monitoring systems.

This study aims to design, implement and evaluate a secure Smart Hostel Management System that integrates both facial recognition and predictive behavioural analytics. The highlights of the technologies involved contribute to the enhanced security and data-driven decision-making in educational institutions. Therefore, this research paper follows a structured approach to investigate the integration of Convolutional Neural Networks (CNN) and Decision Tree algorithms for resident management. Section 2 provides a literature review that discusses the prior research on hostel management systems, facial recognition, and predictive analytics. In Section 3, it explains the methodology that consists of the system architecture, facial recognition implementation and the predictive analytics engine. Section 4 will show the results and discuss the outcome of the research.

*Corresponding author:

E-mail address: Goh Qin Nee <gqinnec@gmail.com>.

<https://doi.org/10.56532/mjsat.v6i3.697>

2785-8901/ © 2026 The Authors. Published by Penteract Technology.

This is an open access article under the CC BY-NC 4.0 license (<https://creativecommons.org/licenses/by-nc/4.0/>).

2. LITERATURE REVIEW

2.1 Hostel Management Systems

Traditionally, hostel management processes are strongly dependent on manual processes. This practice remains prevalent due to its simplicity and accessibility to personnel with limited technological expertise. However, despite these advantages, manual hostel management systems are associated with several limitations that hinder operational efficiency, data accuracy and scalability.

Manual systems allow for the fragmentation of critical records. Student data are scattered across paper documents, physical ledgers and isolated spreadsheets. This approach prevents the necessary unifying of information and hinders the establishment of a single source of truth that is required for effective information handling and retrieval[2]. It is stated that the structural fragmentation of data means the record lacks the fundamental characteristics of a modern effective management system which can guarantee authenticity and consistent usability over time[3].

Manual maintenance of daily entry/exit logs is a time consuming and tedious process for wardens. This inefficiency is best described as requiring an extensive duration of execution and followed by a secondary phase of data entry for report generation[4]. Critically, manual records also suffer from low security and are highly vulnerable to manipulation. Paper-based logs can be erased and overwritten, or the entire record could be falsified by fake data[5]. This vulnerability facilitates malpractices such as proxy attendance, where students can sign in on behalf of their absent peers, rendering the logs to be unreliable[4].

The reliance on manual data transcription represents a significant weakness of traditional paper-based hostel management systems. Research indicates that manual data entry error rates typically range from 1% to 5%, even in academic and research settings. At the lower bound, a 1% error rate implies that a dataset containing 200 records is likely to contain errors affecting multiple entries. Such inaccuracies are often difficult to detect through visual inspection and may alter administrative decisions or statistical analyses derived from the recorded data [6].

The proposed hostel management system mitigates these risks by automating the collection and recording of attendance, entry/exit logs, demerit records and leave request information. Rather than requiring staff to manually transcribe information from paper forms into digital records, data are captured directly through system and stored within the database. This eliminates transcription related errors arising from illegible handwriting, omitted records, duplicate entries and incorrect data transfer. While no information system can completely eliminate data quality issues, the automation of data capture substantially reduces the opportunity for human induced recording errors and improves the consistency and accuracy.

2.2 RFID Systems

RFID (Radio-Frequency Identification) is a type of technology that communicates using electromagnetic waves to exchange data between a terminal and an electronic tag attached to an object like a card. It is mostly used to identify and track[7]. RFID systems were introduced to overcome some of the inefficiencies of traditional hostel management systems. These systems mainly use RFID tags embedded in

cards to record entry or exit automatically when a student passes through a reader at the hostel entrance.

RFID-based systems greatly improved accuracy and security in hostel management. A study conducted showed that using RFID systems for attendance taking is 20 seconds faster and also less susceptible to human error in comparison to manual logging [8]. To further demonstrate this, another study proved that the error from manual systems dropped from 5% to 0.2% by shifting to RFID-based smart locks and remained breach free for the 6 months observed[9].

However, despite these benefits, RFID still has their limitations. The main challenge of RFID-based systems is impersonation[10]. RFID cards can be cloned, allowing unauthorized users to impersonate legitimate cardholders [10]. Therefore, the actual cardholder does not need to be present. Anyone in possession of the card can enter or exit the hostel freely. Furthermore, the reliability of these systems is challenged by reader collisions and uneven tag distributions, which create unbalanced loads across the network. Consequently, simultaneous read attempts can lead to signal interference and blocked tags that severely reduces the overall identification efficiency[11]. The fundamental weakness in this method is their lack of robustness. Its performance is highly sensitive to external variables that are common in a real-world condition. A practical facial recognition system must be able to be robust to image variation caused by lighting conditions, facial expressions or perspectives.

The critical failures of classical algorithms led to the dominance of deep learning models, specifically Convolutional Neural Networks (CNNs). Most modern facial recognition systems often rely on CNN. CNN works by extracting key features on the face like eye position and nose shape[12]. It then converts these features into numerical representations known as facial embeddings which is then stored in a database and then used for comparison during identification or verification.

2.3 Facial Recognition for Security and Monitoring

Facial recognition has become one of the most widely adopted biometric technologies for authentication. It is a type of biometric authentication technology that uses facial features to identify and verify a person's identity. In the context of a hostel, facial recognition ensures that only authorized residents and staff can enter or exit the premises.

Early approaches to facial recognition relied on a classical algorithm that attempted to identify faces using holistic or feature based statistical methods. As an example, Eigenfaces uses the Principal Component Analysis (PCA) to treat face images as a high dimensional vector. PCA acts as feature extraction in face recognition to reduce large dimensionality of data space to smaller dimensionality of feature space when there is strong correlation between observed variables[13].

Unlike traditional systems like manual registers and RFID cards, CNN-based facial recognition systems were a better alternative as they improved accuracy, real time recognition and are resistant to impersonation [14]. Besides that, classical methods often rely on pre-programmed manual feature extraction, while CNNs perform automatic and hierarchical feature learning. They are inspired by the human visual cortex, using multiple layers to learn a progressively complex set of features[15]. This deep learned representation is the key to

their success as it makes them exceptionally robust to the very problems that are present in classical models.

Several CNN architectures have been developed for facial recognition with all of them offering different trade-offs between accuracy and computational efficiency. One of those models is the VGG-Face model that was developed by the Visual Geometry Group at Oxford. It is a deep CNN architecture with 16 layers and is known for its high accuracy. Comparisons between various deep learning models concluded that deeper networks like VGG-Face performed better than shallower networks in unconstrained environments[16]. They generally achieve higher verification rates when faces are not perfectly aligned which is ideal for the hostel environment. In comparison, FaceNet which was developed by Google utilizes a triplet loss function to map face images directly into a Euclidean Space where distances directly correspond to facial similarity[17]. While it does achieve state of the art accuracy when it comes to benchmark datasets like LFW (Labeled Faces in the Wild), it still requires significant computational resources for training and inference. Whereas a lighter weight model like MobileNet is optimized for speed on mobile devices but sacrifices a degree of accuracy compared to VGG-Face.

However, just like RFID- based systems, facial recognition systems also present their own challenges. Facial spoofing is where a fraudster attempts to gain access to a system by impersonating as a valid user[18]. Residents may use photos or videos of someone else to falsify their entry or exit records. Therefore, liveness detection can be applied. Liveness detection is defined as evaluating the movement of the face like eye blinking or prompting the user to perform specific actions like nodding of head or smiling[19]. This ensures that the person is actually present before entry or exit record is taken.

2.4 Predictive Behavioural Analytics

Predictive behaviour analytics is an analytical approach that forecasts human actions in unpredictable settings by integrating process engineering with deep learning techniques to identify anomalies and adjust system responses accordingly[20]. It involves using historical data such as entry and exit logs and demerit scores to identify residents who may be at risk of violating hostel regulations. Modern hostels should apply behavioural monitoring to ensure residents adhere to curfew and rules. In the past, wardens must either review logbooks or CCTV footage in order to track residents' behaviour. This is not only time consuming but also not feasible with hundreds of residents staying in hostels.

With predictive behavioural analytics, machine learning (ML) plays a crucial role. It allows for the system to learn from past data and make future predictions. The application of predictive analytics in higher education has rapidly evolved, mainly focusing on learning analytics to address institutional challenges like student persistence and dropout risk [21, 22]. Learning analytics models focus on predicting key outcomes by analysing log data from learning management systems, course attendance and grades. In the context of a hostel, this translates to analysing movement and disciplinary log data to predict behavioural risk. Studies showed how decision trees algorithms could analyse students' learning profiles and accurately predict their academic performance[23]. Besides

that, Random Forest optimized with grid search to accurately predict student dropout behaviour[24].

While a model like Random Forest might provide higher accuracy for predictions, it obscures the decision process due to their complex multitree voting mechanisms[25]. If an intervention were to happen for the resident, the warden will not be able to explain why that resident is classified as a high risk. Therefore, the prediction will just lead to more questions. Consequently, the Decision Tree emerges as the most suitable choice. Unlike the opaque voting mechanism of a Random Forest, a Decision Tree generates explicit 'if-then' logical rules [26]. By utilizing specific behavioural features, the model can output clear rules, such as, isolating whether a resident was flagged as high risk due to a high frequency of past curfew exits, excessive leave requests or high demerit scores. This evidence based rationale can then be used by a warden to start a supportive conversation and justify actions taken against residents classified as high risk.

3. METHODOLOGY

3.1 System Architecture

The system utilizes a 3-tier architecture built on a WAMP stack (Windows, Apache, MySQL, PHP) for the web server and a Python environment for AI components detailed in Figure 1. The application consists of three interconnected modules:

1. Web Portal: A PHP-based interface for residents, wardens, and admins.
2. Facial Recognition Kiosk: A standalone Python application deployed at entry points.
3. Predictive Analytics Engine: A backend Python service utilizing machine learning to forecast risk.

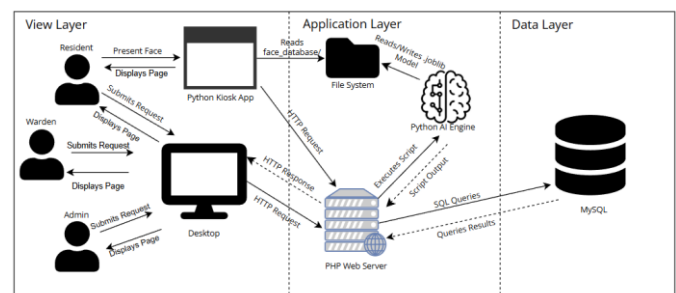


Fig. 1. System Architecture

3.2 Facial Recognition Implementation

The facial recognition kiosk is a standalone Python application designed to operate at hostel entry and exit points. It was developed using Python 3.11 and utilizes the DeepFace framework to interface with deep learning models.

The system initiates a continuous video stream using OpenCV (cv2.VideoCapture). For each frame, it employs the dlib frontal face detector to identify face locations. This method was selected to ensure consistent detection before passing the Region of Interest (ROI) to the recognition engine.

Upon detecting a face, the system uses the VGG-Face model within the DeepFace framework to generate 2,622-dimensional facial embeddings. This embedding size is a direct output of the VGG-Face architecture introduced by

Parkhi et al., where the final fully connected layer of the convolutional neural network produces a fixed length identity feature vector representing facial characteristics in a compact numerical form [27]. This dimensionality is determined by the pretrained model architecture and is not a tuneable parameter within the proposed system.

These embeddings encode high level facial identity information, including structural and textural patterns and are used as discriminative representations for face verification rather than raw pixel data. In the context of a hostel based authentication system, which operates as a closed set recognition environment with a limited number of registered users, this dimensionality is sufficient to achieve reliable identity separation without incurring the computational overhead associated with higher dimensional representations.

These embeddings are compared against the registered user database using cosine similarity. A cosine distance threshold of 0.40 is used to determine high confidence identity verification. This value was empirically selected based on analysis of intra-class (same identity) and inter-class (different identity) similarity distributions within the hostel dataset, where genuine matches consistently exhibited lower distance values and clear separation from impostor comparisons [28, 29].

To improve robustness in borderline cases, a rejection threshold of 0.70 is defined. Distances above this value were consistently associated with non-matching identities during validation experiments and are therefore classified as unknown users. The interval between 0.40 and 0.70 represents an uncertainty region, in which additional verification may be required.

The `DeepFace.find()` method further optimises the comparison process through vectorised NumPy and Pandas operations by enabling efficient batch computation of similarity scores across the registered database. This improves recognition performance by reducing computational overhead compared to sequential pairwise comparison.

3.3 Liveness Detection

Additionally, to mitigate spoofing attacks using photos or videos, a challenge-based liveness detection algorithm was implemented. The system uses the `dlib` 68-point shape predictor to map facial landmarks in real time. If the initial facial match confidence is not absolute, the user is prompted to perform a specific action, either "Blink Eyes" or "Mouth Open".

The system calculates the Eye Aspect Ratio (EAR) to detect blinking. This metric was introduced by Soukupová and Čech and quantifies eye openness based on facial landmark geometry [30]. To filter out noise and normal eye movements, a blink is only registered if the EAR drops below 0.22 for at least 2 consecutive frames. This threshold was determined empirically during system testing by analysing the distribution of EAR values under open eye and closed eye conditions. Observations indicated that EAR values for open eyes consistently remained above approximately 0.25, while values during full eye closure dropped below approximately 0.20. The selected value of 0.22 provides a separation margin between these states, reducing false positives caused by partial closures (e.g., squinting or downward gaze) while maintaining robustness across natural variations in eye shape.

This value is consistent prior EAR-based implementations, including real time blink detection systems derived from Soukupová and Čech, which commonly operate with thresholds around 0.20, as well as subsequent applications in driver drowsiness detection by Rupani et al., where an empirically derived threshold of approximately 0.25 was shown to effectively distinguish between open and closed eye states [30, 31].

Alternatively, the system challenges the user to "Open Mouth". It calculates the Mouth Aspect Ratio (MAR) based on the distance between the inner lip landmarks. The system verifies the action by checking if the MAR exceeds a specific threshold, indicating the user's mouth is open, in order to confirm the user is live and responsive. Access is granted only upon successful completion of either of these challenge. If the user fails these challenges 3 consecutive times, access is directly denied.

3.4 Communication Layer

Furthermore, to ensure a seamless user experience, the kiosk employs an asynchronous fire and forget execution model that can be describe as architectural pattern in modern edge and IoT computing designed to maximize transmission performance by eliminating communication bottlenecks[32]. In this context, fire refers to asynchronously sending the request, while forget refers to proceeding without waiting for or relying on the server's response.

Studies on terminal authentication systems report that standard face recognition processes introduce an average authentication delay of 4.16 seconds per request, caused by the combination of network transmission and remote server processing times[33]. In a linear queuing model, waiting for this server response before updating the user interface creates a significant physical bottleneck. For example, during a peak hour scenario with 150 returning residents, a 4.16 second delay per student accumulates to over 10 minutes of waiting time. Identity and liveness are verified entirely locally on the kiosk. Once identity and liveness are verified, the user interface (UI) immediately updates to display "Access Granted" to the user.

However, attendance logs still need to be synchronized with the central administration database. To achieve this without reintroducing delays, the system separates the workload into parallel threads. Simultaneously, the HTTP POST request containing the log data is sent to the central web server in a background thread (fire). Because the remote server is utilized exclusively for logging and record synchronization rather than real time access approval, the system intentionally does not halt to wait for a success or error response from the server (forget). Therefore, the local kiosk never waits for this remote acknowledgment, preventing network latency from blocking the user interface. This design helps maintain system responsiveness and improves the efficiency of the physical queue during peak usage.

3.5 Predictive Analytics Engine

The predictive analytics module is a backend Python service designed to forecast resident risk levels by analysing behavioural patterns. It utilizes the Decision Tree Classifier from the Scikit-learn library.

Due to the absence of pre-existing historical hostel data, a synthetic data generation strategy was implemented. A script,

train_model.py, utilizes the NumPy library (numpy.random) to generate a dataset of 1,000 hypothetical resident records. This dataset includes randomized variations of three key behavioural features: demerit_score, curfew_violations, and leave_requests. Human behaviour is unpredictable, so simple uniform randomization was avoided. Empirical studies in behavioural science and student discipline consistently indicate that human actions tend to exhibit non-uniform skewed distributions, whereby the majority of individuals remain compliant while a smaller subset accounts for a disproportionate number of infractions[34].

Accordingly, feature specific weighted probability distributions were defined to introduce controlled stochasticity into the synthetic dataset. For instance, in the generation of curfew_violations, a high baseline probability (e.g., 0.8) was assigned to zero violations, with progressively lower probabilities allocated to higher counts (e.g., 0.15 for a single violation and <0.05 for repeated infractions), which produces a right-skewed distribution. Analogous probabilistic structures were applied to other behavioral features, including demerit_score and leave_requests, to ensure that the majority of simulated residents exhibit low or normative behavioral values, while a minority demonstrate elevated risk characteristics.

Since each feature was randomly generated for every student, it introduces natural variation between individuals. Although the dataset is synthetic and does not reflect real-world observations, it provides a reasonable approximation of student behaviour for initial model development.

Furthermore, to create a supervised learning environment, a rule-based labelling function assigns a "High Risk" or "Low Risk" label to each record based on administrative thresholds.

The Decision Tree model is trained on this synthetic dataset to learn the decision boundaries between risk categories. To ensure the model generalizes well to real-world data and avoids overfitting, the tree depth was explicitly constrained using the hyperparameter max_depth=5. This forces the model to learn the most significant patterns rather than creating overly complex specific rules.

Once the training is complete, the model is serialized and saved as a binary file, behavior_model.joblib. This serialization allows the prediction script, predict_model.py, to instantly load the pre-trained state from the disk without needing to retrain the model for every new request. When a warden triggers a prediction via the dashboard, the PHP backend executes the Python inference script, which loads this model, processes the current resident's data and outputs the risk classification in real-time.

To adapt to changing hostel policies, administrators can modify risk thresholds, curfew time and demerit thresholds through the system setting page. Upon saving, the backend script, update_setting_process.php, utilizes PHP's shell_exec() function to trigger the Python training script. This passes the new threshold values as command-line arguments and instantly regenerates and overwrites the serialized model to ensure future predictions reflect the updated rules.

4. RESULT AND DISCUSSION

The primary result of this research is the fully functional Smart Hostel Management System that is deployed on a WAMP stack. The UI was built using the Bootstrap framework, ensuring responsiveness across devices. The system also successfully implements strict Role-Based Access Control (RBAC). The system implements RBAC through the assignment of predefined role identifiers to each authenticated user account. Upon successful login, the user's role (Resident, Warden, or Administrator) is retrieved from the database and stored within the active session. The system then redirects users to their designated portal based on this role assignment. This implementation ensures that users can only access functions relevant to their responsibilities, thereby preventing wardens or residents from modifying administrative system settings and preserving the integrity of critical system configurations.

4.1 Resident Portal

When residents log in, they are presented with a personal dashboard that provides immediate feedback on their standing. It displays real-time counters for demerit points and a history of their recent entry/exit logs. To eliminate paper-based workflows, the portal also includes a digital request management module where residents can submit requests for room changes or overnight leave directly through the system.

4.2 Warden Portal

The warden dashboard features a panel that showcases total active residents, exits recorded today and pending feedback. A key feature of this dashboard is the disciplinary and risk management interface, where wardens can assign demerit points and view the output of the Predictive Analytics Engine. This module integrates the AI risk assessment directly into the workflow and allows for wardens to generate and download PDF reports that flag high risk residents based on behavioural patterns.

4.3 Admin Portal

The admin portal allows administrators to manage the physical infrastructure by creating new room slots and creating new staff accounts. Furthermore, the portal includes a system settings module that controls the AI model's behaviour. Administrators can adjust specific risk thresholds such as the number of demerit points that constitute "High Risk" or the curfew time via a simple form. updates the system's decision parameters, which are then applied to the predictive analytics module. Thus, ensuring the predictive logic evolves alongside changing hostel policies.

4.4 Entry and Exit Kiosk

The entry and exit points are managed by a standalone facial recognition kiosk. In order to prevent queueing during peak hours, the kiosk employs an asynchronous "fire and forget" interface logic. Once the resident's face is authenticated and liveness (blinking or mouth movement) is verified, the UI immediately displays "Access Granted" to the user while the system processes the HTTP logging request in the background. This decoupling ensures that network latency does not impede physical access and provides a smooth entry experience for the residents.

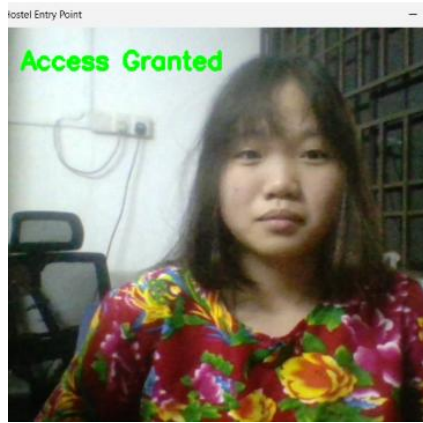


Fig. 2. Entry and Exit Kiosk (Access Granted)



Fig. 3. Entry and Exit Kiosk (Unknown User)



Fig. 4. Entry and Exit Kiosk (Liveness Challenge)

Predictive Behavioural Report

Generated on: 17 Dec 2025, 03:38 AM

AI Prediction Summary

High Risk Residents:	1
Low Risk Residents:	5

Resident Name	Predicted Risk	Confidence	Date
Goh Qin Nee	low	95.35%	16 Dec 2025
Alice Tan	low	95.35%	16 Dec 2025
Jasper Yap	high	88.89%	16 Dec 2025
Aneesa binti Muzammil	low	95.35%	16 Dec 2025
Kavi Pria Moorthy	low	95.35%	16 Dec 2025
Sunyt Singh	low	95.35%	16 Dec 2025

Fig. 5. Sample Predictive Behavioural Report

4.5 Limitations

Despite the successful deployment, the system exhibits certain constraints. Testing revealed that the facial recognition module remains sensitive to unconstrained lighting environments. This is a common occurrence specifically when extreme backlighting is present at entry/exit points, particularly during daylight conditions when subjects stand in front of bright exterior backgrounds. Under such conditions, automatic exposure adjustment mechanisms in imaging pipelines may lead to underexposure of foreground subjects, resulting in reduced visibility of facial regions and loss of discriminative features. Future development will mitigate this by integrating an adaptive exposure control mechanism using OpenCV. Following initial face detection, the facial Region of Interest (ROI) is isolated. The average pixel intensity within this ROI is then evaluated, and if it falls below a predefined visibility threshold, Contrast Limited Adaptive Histogram Equalization (CLAHE) is applied. Unlike global histogram equalization, which can amplify noise and degrade image quality, CLAHE enhances local contrast within the ROI by operating on small image tiles while limiting contrast amplification[35]. This improves the visibility of facial structures without excessively distorting surrounding pixel information.

By normalizing facial luminance prior to feature extraction, this adaptive preprocessing step improves the robustness of downstream computations such as the Eye Aspect Ratio (EAR) and facial embedding generation under varying lighting conditions and ensures consistent logging during all hours of the day.

Additionally, the current predictive analytics engine operates on a binary classification logic (High/Low Risk). While effective for flagging severe cases, this lack of granularity limits the flexibility of administrative responses. Future iterations of the system will extend the Decision Tree classification framework to incorporate a “Moderate Risk” tier.

The Moderate Risk class is designed as an intermediate risk state intended to capture early warning behavioural patterns that do not yet warrant high risk classification but indicate potential escalation. Importantly, the definition of this class is not hardcoded, rather, it is parameterised and configurable by hostel administrators in the system settings module to reflect institutional policy requirements.

In this framework, administrators may define the Moderate Risk category using configurable thresholds such as accumulated demerit scores, frequency of curfew violations or excessive leave request patterns similar to the original High Risk and Low Risk settings. This design allows the system to adapt to different institutional disciplinary policies while maintaining a consistent underlying classification structure. This Moderate Risk tier follows the same underlying decision framework as the existing High and Low Risk categories but introduces an additional intermediate layer to improve classification granularity and enable earlier intervention, thereby supporting more nuanced administrative decision making.

5. CONCLUSION

In conclusion, this research introduced a new level of security and intelligence in a hostel environment. The facial recognition entry and exit points eliminated the need for physical logbooks while the implementation of liveness checks mitigates the risk of spoofing. Furthermore, the introduction of the predictive analytics engine marks a step towards accountability towards one's misconduct. Wardens can now not only make informed decisions and take early preventive actions against predicted high risk residents; residents will also have to take accountability for their actions.

While the system operates with certain constraints, it stands as a fully functional and scalable system. It is proof that automating routine tasks and applying intelligent analytics to a system can significantly improve both the operational efficiency and security of a system.

REFERENCES

- [1] A. Walker, "Malaysia on track to attract 250,000 foreign students by 2025," *GSL Global*, February 14, 2025. url: <https://gslglobal.com/2025/02/14/malaysia-on-track-to-attract-250000-foreign-students-by-2025/>
- [2] PowerSchool, "The problem with using spreadsheets for K-12 data analytics," January 2024, url: <https://www.powerschool.com/blog/the-problem-with-using-spreadsheets-for-k-12-data-analytics/#:~:text=But%20a%20spreadsheet%20is%20limited,cells%2C%20rows%2C%20and%20columns.>
- [3] M. Rebuglio, F. M. Ottaviani, and A. D. Marco, "Recordkeeping for project management information system in public procurement: An action research," *Records Management Journal*, February 2025, doi: <https://doi.org/10.1108/RMJ-10-2023-0052>.
- [4] M. Al-Dabbagh, "A review of students attendance management systems," *Eurasian Journal of Science and Engineering*, vol. 10, no. 3, pp. 136–145, April 2025, doi: <https://doi.org/10.23918/eajse.v10i3p13>.
- [5] M. S. I. Shovon, "Security and Privacy Issues Related to Electronic Documents based on the Public Perception," *Journal of Information Education Research*, vol. 5, no. 4, pp. 1678–1686, December 2025, Available: <https://jier.org/index.php/journal/article/view/3954>
- [6] K. A. Barchard and L. A. Pace, "Preventing human error: The impact of data entry methods on data accuracy and statistical results," *Comput. Hum. Behav.*, vol. 27, no. 5, pp. 1834–1839, September 2011, doi: <https://doi.org/10.1016/j.chb.2011.04.004>.
- [7] R. Rao, "Radio Frequency Identification," *International Journal of Innovative Research in Advanced Engineering*, vol. 09, no. 12, pp. 489–492, December 2022.
- [8] M. I. M. Rozlan et al., "RFID-based attendance monitoring system with LED authentication," in *Proceeding 2023 IEEE International Conference on Automatic Control and Intelligent System (I2CACIS)*, June 2023, pp. 85–90, doi: <https://doi.org/10.1109/I2CACIS57635.2023.10193394>.
- [9] M. Imran, M. Fikry, and S. Retno, "Enhancing academic security with RFID-based smart locks and real-time attendance tracking system," in *Proceedings of Malikussaleh International Conference on Multidisciplinary Studies (MICoMS)*, vol. 4, December 2024, doi: <https://doi.org/10.29103/micoms.v4i.950>.
- [10] H. Pereira, R. Carreira, P. Pinto, and S. I. Lopes, "Hacking the RFID-based authentication system of a university campus on a budget," in *Proceedings Iberian Conference on Information Systems and Technologies (CISTI)*, 2020, pp. 1–5, doi: <https://doi.org/10.23919/CISTI49556.2020.9140943>.
- [11] Y. Wang et al., "A Protocol Stack for Large-Scale RFID Systems: Mitigating Reader and Tag Collisions," *IEEE Internet of Things Journal*, vol. 11, no. 19, pp. 30455–30468, May 2024, doi: <https://doi.org/10.1109/IJOT.2024.3398650>.
- [12] A. A. Tamimi, O. N. Al-Allaf, and M. A. Alia, "Eigen faces and principle component analysis for face recognition systems: A comparative study," *International Journal of Computer and Technology*, vol. 14, no. 4, pp. 5650–5660, February 2015, doi: <https://doi.org/10.24297/ijct.v14i4.1967>.
- [13] A. K. Singh, S. Kumar, B. Kumar, and A. S. Chauhan, "Face Recognition Using Machine Learning," in *2023 International Conference on Sustainable Emerging Innovations in Engineering and Technology (ICSEIET)*, pp. 361–364, September 2023, doi: <https://doi.org/10.1109/icseiet58677.2023.10303390>.
- [14] V. K. Kulamala et al., "Non-intrusive facial recording system with convolution neural network (cnn) for attendance assimilation," May 2024, pp. 130–150, doi: <https://doi.org/10.58532/v3baai7p1ch9>.
- [15] L. Chen et al., "Review of image classification algorithms based on convolutional neural networks," *Remote Sens.*, vol. 13, no. 22, p. 4712, November 2021, doi: <https://doi.org/10.3390/rs13224712>.
- [16] D. Saez-Trigueros, L. Meng, and M. Hartnett, "Face recognition: From traditional to deep learning methods," *arXiv*, October 2018, doi: <https://doi.org/10.48550/arXiv.1811.00116>.
- [17] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," in *Proceedings IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015, pp. 815–823, doi: <https://doi.org/10.1109/CVPR.2015.7298682>.
- [18] E. Solomon and K. J. Cios, "FASS: Face anti-spoofing system using image quality features and deep learning," *Electronics*, vol. 12, no. 10, p. 2199, May 2023, doi: <https://doi.org/10.3390/electronics12102199>.
- [19] C. Widjaya and A. Wicaksana, "Liveness detection with randomized challenge-response for face recognition anti-spoofing," *International Journal of Innovative Computing, Information and Control*, vol. 19, no. 2, pp. 419–430, April 2023, doi: <https://doi.org/10.24507/ijicic.19.02.419>.
- [20] A. Koschmider and S. Speidel, "Predictive behavior analysis for smart environments," *Emisa Forum*, vol. 36, pp. 79–82, October 2016.
- [21] D. Abbott, *Applied predictive analytics: Principles and techniques for the professional data analyst*. Wiley, 2014.
- [22] P. Lourens and D. Bleazard, "Applying predictive analytics in identifying students at risk: A case study," *South African Journal of Higher Education*, vol. 30, no. 3, June 2016, pp. 129–142, doi: <https://doi.org/10.20853/30-2-583>.
- [23] V. Matzavela and E. Alepis, "Decision tree learning through a predictive model for student academic performance in intelligent m-learning environments," *Computers and Education: Artificial Intelligence*, vol. 2, Art. no. 100035, January 2021, doi: <https://doi.org/10.1016/j.caeai.2021.100035>.
- [24] D. Kumar, A. Kothiyal, R. Kumar, C. Hemantha and R. Maranan, "Random Forest approach optimized by the Grid Search process for predicting the dropout students," in *2024 International Conference on Innovations and Challenges in Emerging Technologies (ICICET)*, pp. 1–6, Jun. 2024, doi: <https://doi.org/10.1109/ICICET59348.2024.10616372>.
- [25] A. Assis, J. Dantas, and E. Andrade, "The performance-interpretability trade-off: a comparative study of machine learning models," *Journal of Reliable Intelligent Environments*, vol. 11, no. 1, Art. no. 1, December 2024, doi: <https://doi.org/10.1007/s40860-024-00240-0>.
- [26] A. Rai, "Explainable AI: from black box to glass box," *Journal of the Academy of Marketing Science*, December 2020, doi: <https://doi.org/10.1007/s11747-019-00710-5>.

- [27] O. M. Parkhi, A. Vedaldi, and A. Zisserman, "Deep Face Recognition," in *Proc. British Machine Vision Conference (BMVC)*, Swansea, UK, 2015, doi: <https://doi.org/10.5244/C.29.41>.
- [28] A. K. Jain, A. Ross, and S. Prabhakar, "An introduction to biometric recognition," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 14, no. 1, pp. 4–20, Jan. 2004, doi: <https://doi.org/10.1109/TCSVT.2003.818349>.
- [29] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 815–823, doi: <https://doi.org/10.1109/CVPR.2015.7298682>.
- [30] T. Soukupová and J. Čech, "Real-time eye blink detection using facial landmarks," in *Proc. 21st Computer Vision Winter Workshop*, Rimske Toplice, Slovenia, Feb. 2016, pp. 1–8.
- [31] V. S. K. Rupani, V. V. S. Thushar, and K. Tejith, "Real-time drowsiness detection using eye aspect ratio and facial landmark detection," *arXiv*, arXiv:2408.05836, Aug. 2024, doi: <https://doi.org/10.48550/arXiv.2408.05836>.
- [32] A. Almheiri and Z. Maamar, "IoT Protocols – MQTT versus CoAP," in *Proceedings of the 4th International Conference on Networking, Information Systems & Security (NISS 2021)*, Association for Computing Machinery, 2021, doi: <https://doi.org/10.1145/3454127.3456594>.
- [33] Q. Li, H. Xiang, W. Cui, K. Liu, and D. Li, "A Multi-Factor Authentication Method for Power Grid Terminals Based on Edge Computing Paradigm," *IEEE Access*, vol. 13, pp. 166062–166077, 2025, doi: <https://doi.org/10.1109/ACCESS.2025.3609692>.
- [34] S. Mondal et al., "Most people obey arbitrary rules even when it's not in their interest to do so," *Nature Human Behaviour*, vol. 9, pp. 1342–1354, 2025.
- [35] I. M. Mohammed and N. A. Mat Isa, "Contrast Limited Adaptive Local Histogram Equalization Method for Poor Contrast Image Enhancement," *IEEE Access*, vol. 13, pp. 62600–62632, 2025, doi: <https://doi.org/10.1109/ACCESS.2025.3558506>.