Robust Deep Learning-based Secret Key Generation in Dynamic LiFi Networks Against Concept Drift

Elmahedi Mahalal*, Muhammad Ismail*, Zi-Yang Wu†, Mostafa M. Fouda‡§,
Zubair Md Fadlullah¶, and Nei Kato$^\|$
*Department of Computer Science, Tennessee Technological University, Cookeville, TN, USA,
†College of Information Science and Engineering, Northeastern University, Shenyang, China,
‡Department of Electrical and Computer Engineering, Idaho State University, Pocatello, ID, USA,
§Center for Advanced Energy Studies (CAES), Idaho Falls, ID, USA,
¶Department of Computer Science, Western University, London, ON, Canada,
$^\|$Graduate School of Information Sciences, Tohoku University, Sendai, Japan,
Emails: {emahalal42, mismail}@tntech.edu, wuziyang@ise.neu.edu.cn, mfouda@ieee.org,
zfadlullah@ieee.org, and kato@it.is.tohoku.ac.jp

Abstract—**This paper explores secret key generation in 5G and beyond LiFi networks using visible light in the downlink and infrared in the uplink. Unlike the existing works, we focus on a realistic indoor environment with multi-user mobility. Given inaccuracies in high-frequency channel models, we introduce the first deep learning model that combines the channel probing and quantization phases to generate initial secret keys with a minimal key disagreement rate (KDR) of 16% between the uplink and downlink, leading to a key generation rate (KGR) of 79 bits/s after information reconciliation. We show that LiFi channel statistics suffer from concept drifts with user density changes in the room. This increases the KDR by $28\% - 44\%$ and the generated keys fail to pass the NIST randomness tests. As a countermeasure, we introduce a voting ensemble model that mitigates concept drifts, maintaining a stable 16% KDR, 79 bits/s KGR, and passing NIST tests, despite the varying user densities.**

Index Terms—**Wireless secret key, LiFi, VLC, concept drift, indoor mobility, 5G+ networks, ensemble, deep learning.**

I. Introduction

With the growth of wireless devices and networks, data encryption in wireless communications has become crucial. Cryptographic methods typically rely on pre-shared keys, which may not always be available [1]. Instead, wireless secret key generation derives the secret shared key from the wireless channel. This key should conform to the NIST randomness standards [2] and is used for encrypting and decrypting communications between the user and the access point (AP).

To create a wireless secret key, both parties (user and AP) probe and quantize channels in the uplink (UL) and downlink (DL), producing preliminary keys. Information reconciliation then ensures identical keys by removing differing bits. While the literature often assumes channel reciprocity, implying preliminary key similarity without reconciliation, this is not always true, especially in 5G and beyond (5G+) networks. For instance, LiFi uses visible light downlinks and infrared uplinks, with different characteristics, making the reciprocity assumption invalid. Thus, to maximize the key generation rate (KGR), an efficient process is required to attain a low key disagreement rate (KDR) between the UL and DL before reconciliation.

Given that 80% of data traffic originates indoors [3], our study focuses on indoor secret key generation. The current research has not explored thoroughly this topic in LiFi networks, known for their sensitive, non-ideally reciprocal, high-frequency channels. Recent findings suggest no general channel models for LiFi indoor networks [4], as they are influenced by room layout and interactions among users and objects. Consequently, using a model-based approach to optimize the KDR in preliminary keys is not possible. Instead, data-driven methods should be adopted by employing machine learning on channel gain data to achieve low KDR in the preliminary keys.

Recent efforts have explored data-driven approaches and deep learning techniques for wireless secret key generation, but none combined channel probing and quantization [5]–[7], two main steps to reduce the mismatch between the UL and DL. Moreover, the effects of multi-user mobility and varying user densities have not been examined. As the number of mobile users in a room changes, so do channel blockage and outage rates, especially in LiFi channels, which are sensitive to blockages. This raises the following open questions: (a) How do user mobility and different densities influence a data-driven secret key generation method? (b) How can we create a robust data-driven secret key generation with minimal KDR that meets NIST randomness tests, regardless of user mobility and varying densities?

A. Related Work

Most existing studies focus on wireless secret key generation in radio channels, such as [8], or in mmWave channels, such as [9] and [10]. The prevalent assumption in these works is ideal channel reciprocity. However, recent research has considered non-ideal channel reciprocity, using deep learning to establish secret key generation frameworks. For example, [5] introduced a deep learning framework for time division duplex systems, accounting for asynchronous channel information and hardware

This work was supported by NSF award #2138234. Dataset generation was supported by the TNTech HPC cluster funded by NSF award #2127188.

differences, with two auto-encoders that improved the KDR. The study in [7] proposed a framework utilizing randomized pilots and deep learning to boost randomness and combat potential attacks. Additionally, [11] used a machine learning strategy for predicting quantization levels achieving a 98% accuracy rate. Also, [12] employed neural networks to deduce wireless channel features for key generation, achieving high agreement rates and hardware adaptability. Finally, [6] proposed a deep learning-based key generation method, focusing on mapping features across diverse frequency bands, typical in non-reciprocal frequency-division duplexing systems.

Limitations: To the best of our knowledge, no existing study presents a deep learning model that integrates channel probing and quantization to tackle channel non-ideal reciprocity and minimize the KDR. While [11] uses deep learning solely for the quantization phase, works like [5]–[7] separate channel feature extraction from the quantization process. Further, all the existing studies focus on radio and mmWave channels. Only [13] explores secret key generation in LiFi networks within mobile setups, but without minimizing the KDR, resulting in a 40% KDR. LiFi networks, essential in 5G+, differ from radio and mmWave channels. They display non-ideal reciprocal channels and are sensitive to user movement, leading to blockages that can cause outages and affect the channel impulse response. The literature has not fully addressed the implications of user mobility and varying densities on deep learning-based secret key generation in 5G+ networks.

B. *Contribution*

To fill in the research gap, we carried out the following:

- We present the first deep learning framework that combines channel probing and quantization to minimize the KDR of the preliminary keys in *indoor LiFi networks* with *multi-user mobility*. Our practical framework accounts for indoor human mobility on two timescales, reflecting macro and micro-mobility patterns. Mobility traces are used to generate the channel impulse response (CIR), accounting for blockages and transmitter-receiver misalignments due to movement. The CIR data is used to train a deep long-short-term-memory (LSTM) recurrent neural network (RNN) at the AP to predict quantized preliminary keys closely matching those generated at the user equipment (UE), thereby minimizing the KDR in preliminary keys.
- For the first time, we assess the effects of *multi-user mobility* and various *user densities* on secret key generation in indoor LiFi networks using three metrics, KDR, KGR, and the NIST randomness tests [2]. Our LSTM-RNN model, trained with CIR from a fixed user density, achieves a 16% KDR, which represents a 24% improvement over [13] and passes the NIST tests. However, changing user density increases the KDR to 44% and keys fail the NIST tests, even when training with CIR gathered from all density scenarios. This stems from the concept drift effect, caused by shifts in outage events and CIR probability distributions when user density changes.
- To counter concept drifts, we propose an ensemble-based LSTM-RNN model that provides the optimal quantized keys at the AP based on the input CIR data. This ensures consistent key generation performance irrespective of user mobility or user density. Our results show this model keeps KDR at 16% and the keys pass the NIST randomness tests.

The rest of this paper is structured as follows. Section II describes the system model. Section III introduces the deep learning-based key generation framework and assesses its performance with varying user densities. Section IV details and evaluates the ensemble model. Conclusions are in Section V.

II. SYSTEM MODEL

This section presents the indoor setup, the human mobility model, and the channel model.

A. *Indoor Setup*

We consider an office room with dimensions 5 m × 5 m × 5 m. The room has nine desks measuring 1 m × 0.75 m × 1.3 m. Four LiFi APs cover the room, which are uniformly distributed across the ceiling, as shown in Fig. 1. The human body is approximated as a cuboid with dimensions 1.8 m × 0.2 m × 0.45 m. The mass of the human body is assumed to be 70 Kg with a peak walking speed of 2.1 m/sec. and a maximum acceleration of 1 m/sec^2. The indoor mobility sample interval is 100 milli-sec. All the surfaces in the room and the human body are treated as reflectors and blockers of the line-of-sight (LoS) channels in the UL and DL channels.

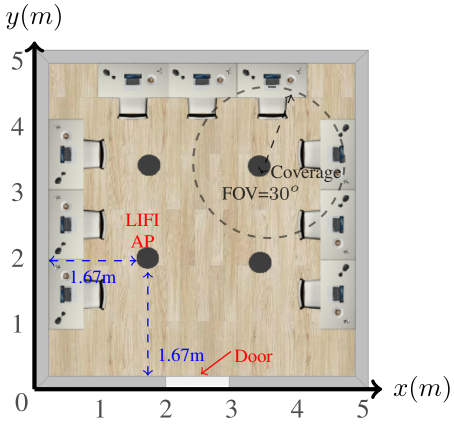


Fig. 1: Office room setup.

B. *Indoor Human Mobility Model*

We adopt the indoor human mobility model described in [4]. This model realistically reflects indoor human movements on two timescales (a) the macro scale, which specifies the time instant that human moves to the next destination point and specifies that destination point, while (b) the micro-scale captures the details of mobility as human moves from one point to another. The macro scale is represented by a semi-Markov renewal process that reflects return regularity and bounded Lévy-walk. The micro-scale implements the shortest path, steering behavior, and orientation of mobile devices. The mobility model is summarized in the upper section of Fig. 2. The synthetic mobility traces generated by this model were validated in [4] using real measurements collected with the Phyphox application. Further details about the mobility model parameters can be found in [4].

C. Channel Model

The UL channel is in the visible light band and the DL channel is in the infrared band. The LoS CIR at a specific time, t, is provided by [4]

$$h(t) = \begin{cases} \dfrac{A}{d_0^2(t)} \dfrac{(m+1)}{2\pi} \cos^m \psi(t) \cos\theta(t) \times R\left(\theta(t)\right), \\ \qquad \text{if } 0 \leq \theta(t) \leq \Psi, \\ 0, \quad \text{if } \theta(t) > \Psi \text{ or ray is blocked}, \end{cases} \tag{1}$$

where A denotes the detector's area (100 mm^2), m is the mode number (Lambert sources are assumed), ψ stands for the angle of irradiance, Ψ is the receiver's field-of-view (FoV = 30^o), $R(\theta)$ signifies the general transmission response of the optical system (assumed 1 for simplicity), $d_0(t) = ||\mathbf{p}(t) - \mathbf{p}_{\text{AP}}(v)||$, $\mathbf{p}_{\text{AP}}(v)$ denotes the location of AP $v = \{1, \ldots, 4\}$, and $\mathbf{p}(t)$ represents the user position.

III. Deep Learning-based Framework for Secret Key Generation In LiFi Networks

Fig. 2 outlines our approach to generating wireless secret keys in mobile LiFi networks. The process begins with creating indoor human mobility traces as described in Section II. B. Then, the blockage is determined using ray tracing by checking for signal intersections with room objects. We also verify if the received signal is within the receiver's FoV. Then, the CIR is computed as per (1). The secret key is formed through channel probing, quantization, and information reconciliation. After introducing these steps, we will detail our deep-learning method to minimize the KDR in preliminary keys caused by channel non-ideal reciprocity.

A. Channel Probing, Quantization, and Reconciliation

We discuss herein the basic steps to generate a wireless secret key between the user and the AP.

1) Channel Probing: Initially, the user and AP engage in a two-way exchange of request and response probing frames over some duration. After a request frame is received, the receiver responds with a reply frame. A constant interval, τ, is assumed between any two consecutive requests (or reply) probing frames, resulting in a channel probing rate of $1/\tau$. By the end of the channel probing process, the user and AP have accumulated a set of N pairs of channel measurements. The estimated channel gains in the DL and UL are

$$\begin{cases} \mathbf{H}_{\text{DL}} & = [h_{\text{DL}}(1), h_{\text{DL}}(2), \ldots, h_{\text{DL}}(N)]^{\text{T}}, \\ \mathbf{H}_{\text{UL}} & = [h_{\text{UL}}(1), h_{\text{UL}}(2), \ldots, h_{\text{UL}}(N)]^{\text{T}}, \end{cases} \tag{2}$$

where T denotes the transpose and $h(n)$ at discrete instances $1 \leq n \leq N$ is the channel gain (CIR) estimate. In this paper, perfect estimation is assumed as a first step of research.

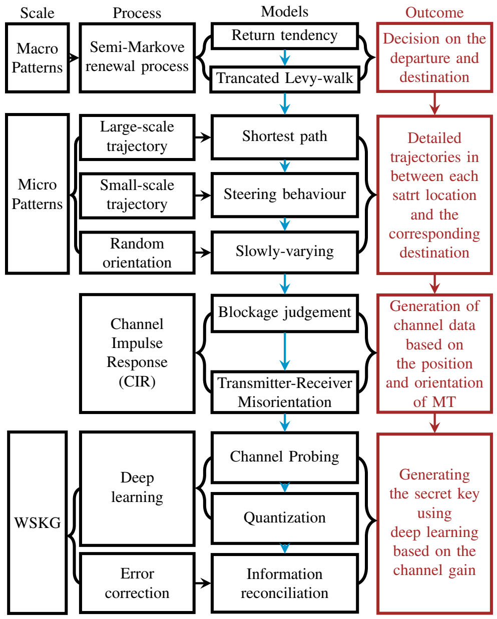

Scale
Process
Models
Outcome
Macro Patterns
Semi-Markove renewal process
Return tendency
Trancated Levy-walk
Decision on the departure and destination
Micro Patterns
Large-scale trajectory
Small-scale trajectory
Random orientation
Shortest path
Steering behaviour
Slowly-varying
Detailed trajectories in between each satrt location and the corresponding destination
Channel Impulse Response (CIR)
Blockage judgement
Transmitter-Receiver Misorientation
Generation of channel data based on the position and orientation of MT
WSKG
Deep learning
Error correction
Channel Probing
Quantization
Information reconciliation
Generating the secret key using deep learning based on the channel gain

Fig. 2: Framework for deep learning-based wireless secret key generation (WSKG) in dynamic multi-user indoor LiFi.

2) Quantization: This paper employs a cumulative distribution function (CDF)-based quantization, where quantization thresholds are set according to the CIR data's CDF. This ensures a balanced output of 1s and 0s, essential for the NIST randomness tests. The approach allows for multi-bit quantization using more levels. Using a Gray code ensures similar data samples yield closely related binary strings with a single bit difference. We chose the CDF-based method since the CIR distribution changes with varying user densities. Algorithm 1 details the CDF-based quantizer, taking in the CIR estimate and desired quantization level to output a preliminary key sequence. It calculates thresholds from the CIR's CDF and assigns Gray codes accordingly. Traditionally, both the UE and AP employ Algorithm 1 to generate preliminary keys using received data.

3) Information Reconciliation: This step produces symmetric keys at the UE and AP by aligning nonidentical bits in the preliminary keys, $\tilde{K}_{\text{UL}}$ and $\tilde{K}_{\text{DL}}$, to form the final key K. Our goal is to minimize the KDR due to channel non-ideal reciprocity before information reconciliation, ensuring a high KGR. For reconciliation, the algorithm in [14] is adopted.

During the probing, quantization, and reconciliation, UE and AP communicate over public channels, risking eavesdropper interception. In this process, the eavesdropper should have the same correlated channel as the legitimate users to obtain similar quantization results. While privacy amplification is typically applied after reconciliation, for effective eavesdropping, the attacker must maintain a distance of $\lambda/2$ (with λ representing the wavelength). Given the challenge of maintaining this distance in LiFi due to the nanometer-scale λ, this paper omits privacy

Algorithm 1 CDF-Based Quantization

```
procedure CDF_QUANTIZATION($\mathbf{H}$, $V$)    ▷ CIR Estimate, Quantization level
    $F(h) \leftarrow \Pr(\mathbf{H} < h)$    ▷ CDF
    $\eta_0, \eta_{2^V} \leftarrow -\infty, \infty$
    for $j \leftarrow 1$ to $2^V - 1$ do
        $\eta_j \leftarrow F^{-1}(j/2^V)$
    end for
    Construct Gray codes $b_j$ for intervals $[\eta_{j-1}, \eta_j]$
    for $n \leftarrow 1$ to $N$ do
        if $\eta_{j-1} \le h(n) < \eta_j$ then
            $\tilde{K}(n, V) \leftarrow b_j$
        end if
    end for
end procedure
```

amplification considerations.

B. Proposed Deep Learning-based Key Generation Strategy

In LiFi networks, UL (infrared) and DL (visible light) channels are non-ideally reciprocal. Optimizing quantization thresholds at the AP and UE is essential for closely matched initial keys, preventing high KDR and reduced KGR after reconciliation. Due to the lack of general LiFi channel models [4], our proposed solution is data-driven, employing a deep learning-based strategy at the AP to minimize the KDR. An LSTM-RNN model is trained with normalized UL CIR ($\bar{\mathbf{H}}_{\text{UL}}$) as input, generating a Gray code output similar to that adopted at the UE. The model minimizes the categorical cross-entropy to minimize the KDR by learning the relation between $\mathbf{H}_{\text{UL}}$ (used at AP to generate $\tilde{K}_{\text{DL}}$) and $\tilde{K}_{\text{UL}}$ (generated at the UE). During tests, the model generates an AP preliminary key closely matching the UE's preliminary key given the UL CIR.

1) Dataset Generation: The framework in Fig. 2 is used to create the UL and DL CIR dataset, processed using the Tennessee Technological University's HPC cluster. Indoor mobility traces were crafted as per Section II.C. User and AP locations then determine potential blockages and coverage, with blockage judgment details in [4]. UL and DL CIRs are derived using parameters from [4] for infrared (UL) and visible light (DL). CIR data is then aggregated for different user densities, namely, $1, 3, 6,$, and 8 users. For each density, 1000 traces per user are generated and processed. This results in time-series data for each user and AP as in (2). Before LSTM-RNN training, data undergoes pre-processing to extract features and labels. The subsequent sections will discuss these details.

2) Dataset Pre-processing: As aforementioned, the LSTM-RNN model is trained and deployed at the AP such that it optimally quantizes $\mathbf{H}_{\text{UL}}$ to produce a $\tilde{K}_{\text{DL}}$ of minimum KDR with $\tilde{K}_{\text{UL}}$. Toward this goal, the following pre-processing steps are taken to define the input features and the output class.

Normalizing Input Features: To allow fast convergence of the LSTM-RNN model training, the UL CIR data is normalized. This is achieved by defining $h_{\text{UL}}^{\max} = \max(\mathbf{H}_{\text{UL}})$ and $h_{\text{UL}}^{\min} = \min(\mathbf{H}_{\text{UL}})$, then, calculating the input features $\mathbf{X} = [x(1), x(2), ..., x(N)]$

$$x(n) = \frac{h_{\text{UL}}(n) - h_{\text{UL}}^{\min}}{h_{\text{UL}}^{\max} - h_{\text{UL}}^{\min}}. \tag{3}$$

Defining Output Labels/Classes: To minimize the KDR, we first find the quantization (Gray code) outcome at the UE given the downlink CIR $\mathbf{H}_{\text{DL}}$. This is done by calling the procedure CDF_QUANTIZATION($\mathbf{H}_{\text{DL}}, V$) in Algorithm 1. Hence, at each time instance n, we have the corresponding Gray code outcome $\tilde{K}_{\text{UL}}(n)$. As the Gray code represents binary outcomes, we find its decimal equivalent and use it as the model's output label/class $y(n)$ at time n. In total, we have 2^V thresholds mapped to 2^V different Gray codes. Hence, we have 2^V classes/labels. The labels/classes corresponding to the input features are denoted by $\mathbf{Y}$.

3) LSTM-RNN Model Training and Optimization: The model utilizes LSTM-RNN for time series data in our dataset while addressing gradient vanishing and exploding issues. The model is trained on examples of $(\mathbf{X}, \mathbf{Y})$ with a $3:1$ split using backpropagation-through-time (BPTT) that minimizes categorical cross-entropy, which is equivalent to minimizing the KDR. Hyper-parameters are refined via grid search and the optimal configuration is found as two hidden layers; first with 56 LSTM cells ($\tanh$ activation) and second with 72 (Sigmoid activation). The output layer has 2^V neurons with a Softmax activation. Additional parameters include a learning rate of 0.001, Adam optimizer, batch size 10, and 1000 epochs.

For testing, the normalized UL CIR is fed to the model as input features, and the UL preliminary keys $\tilde{K}_{\text{UL}}$ are predicted as the model's output, which is then used as $\tilde{K}_{\text{DL}}$. Then, we evaluate the performance of the model based on the following:

- KDR: This is defined as the ratio of the number of mismatched bits in the DL and UL preliminary keys $\tilde{K}_{\text{DL}}$ (generated using the LSTM-RNN model deployed at the AP) and $\tilde{K}_{\text{UL}}$ (generated using Algorithm 1 at the UE), respectively. Let the length of the bits in the preliminary keys be S. Then, the KDR is described as

$$\text{KDR} = \frac{\sum_{s=1}^{S} |\tilde{K}_{\text{UL}}(s) - \tilde{K}_{\text{DL}}(s)|}{S}. \tag{4}$$

- KGR: After information reconciliation, the AP and UE establish a uniform key K with a KDR that tends to be zero. The KGR, measured in bits per second, indicates the key generation/update speed. For example, the AES algorithm needs a KGR of 0.1 bit per second [8].
- NIST Randomness Tests: These are described in [2] and assess key randomness. A key with a p-value ≥ 0.01 in these tests exhibits high randomness. The tests include: *monobit*, *frequency within the block*, *longest run of ones*, *binary matrix rank*, *discrete Fourier transform*, *non-overlapping template matching*, *approximate entropy*, *cumulative sums*, and *random excursion*.

4) Performance Evaluation Results: We evaluated the generalization of the LSTM-RNN through five models. Models $\mathbf{M}_1$, $\mathbf{M}_3$, $\mathbf{M}_6$, and $\mathbf{M}_8$ were trained for scenarios with 1, 3, 6, and 8 mobile users, respectively. A fifth model, $\mathbf{M}_\mathrm{G}$, was trained using data from all user densities. When tested with the same user density they were trained on, models $\mathbf{M}_1 - \mathbf{M}_8$ achieved the lowest KDRs of $15-16\%$, as indicated by the bold diagonal in Table I. However, testing on different user densities resulted in a KDR increase to $37-44\%$. The $\mathbf{M}_\mathrm{G}$ model had a consistent KDR of 28%, still $12-13\%$ higher than the $\mathbf{M}_1 - \mathbf{M}_8$ models tested in matching conditions. This highlights the models' sensitivity to user density.

TABLE I: KDR for different models and test conditions.

Model/Test	1 UE	3 UEs	6 UEs	8 UEs
$\mathbf{M}_1$	**0.1495**	0.4155	0.4226	0.4301
$\mathbf{M}_3$	0.4222	**0.1562**	0.3976	0.4013
$\mathbf{M}_6$	0.4220	0.3872	**0.1577**	0.3923
$\mathbf{M}_8$	0.4401	0.3940	0.3731	**0.1601**
$\mathbf{M}_\mathrm{G}$	0.2897	0.2899	0.2875	0.2889

To assess key randomness using the mentioned models, NIST randomness tests were conducted (Table II). Models $\mathbf{M}_1$ to $\mathbf{M}_8$ passed these tests when trained and tested with the same user density. However, discrepancies emerged in tests with varying user densities. For instance, $\mathbf{M}_1$ struggled in the "overlapping template matching" test for 3 to 8 users, as well as in the "random excursion" test for 8 users. Similarly, $\mathbf{M}_\mathrm{G}$ faced challenges, failing the "overlapping template matching" (with 1 user), "block frequency" (3 users), "random excursions" (6 users), and "frequency within a block" (8 users) tests.

TABLE II: NIST tests for different models and test conditions.

Model/Test	1 UE	3 UEs	6 UEs	8 UEs
$\mathbf{M}_1$	**Passed**	Failed	Failed	Failed
$\mathbf{M}_3$	Failed	**Passed**	Failed	Failed
$\mathbf{M}_6$	Failed	Failed	**Passed**	Failed
$\mathbf{M}_8$	Failed	Failed	Failed	**Passed**
$\mathbf{M}_\mathrm{G}$	Failed	Failed	Failed	Failed

In conclusion, models exhibit limited generalization ability. While models $\mathbf{M}_1 - \mathbf{M}_8$ perform well under fixed user density, they falter with differing user densities, yielding high KDR and failing some NIST randomness tests. Similarly for the $\mathbf{M}_\mathrm{G}$ model. This is attributed to changing CIR and outage distributions with user density, as shown in Figs. $3-5$.

Fig. 3 illustrates varying CIR with user density. From Fig. 4, two key points arise: (a) different user densities lead to varied outage rate distributions, with increased users causing more blockages; (b) UL and DL channels possess distinct outage rates. These factors affect the CIR distribution as seen in Fig. 5. Notably, with more users, the likelihood of high CIR values decreases due to more outages, a phenomenon termed "concept drift". Hence, models trained for a specific user density struggle to generalize for other densities. This is further complicated by the distinct CIR distributions for UL and DL, underscoring the channel non-ideal reciprocity in LiFi networks.

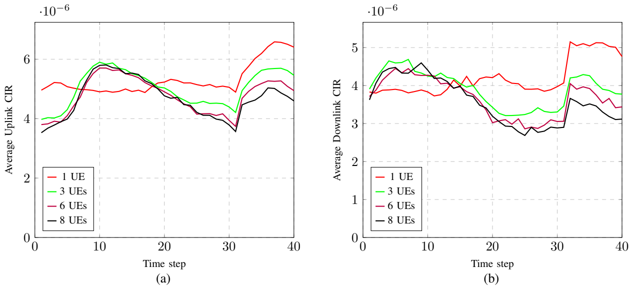


Fig. 3: Time evolution of average CIR for $1,000$ mobility traces during wandering: (a) UL and (b) DL channels.

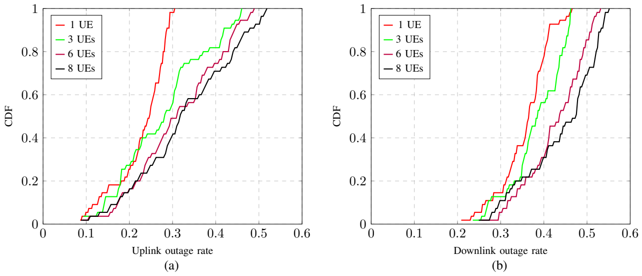


Fig. 4: Cumulative density function for (a) UL and (b) DL outage rate.

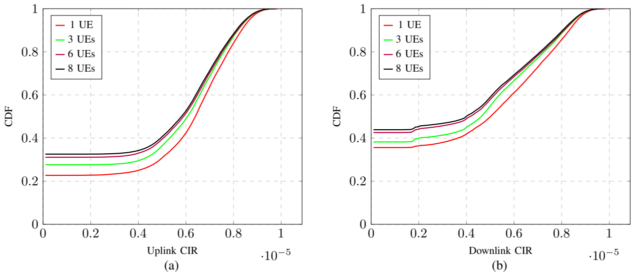


Fig. 5: CDF for (a) UL and (b) DL CIR.

In summary, LiFi networks show (a) non-ideal reciprocity between UL and DL channels, and (b) concept drift in CIR and outage rates with varying user density. These shifts account for the models' weak generalization. The next section introduces an ensemble learning approach to enhance generalization.

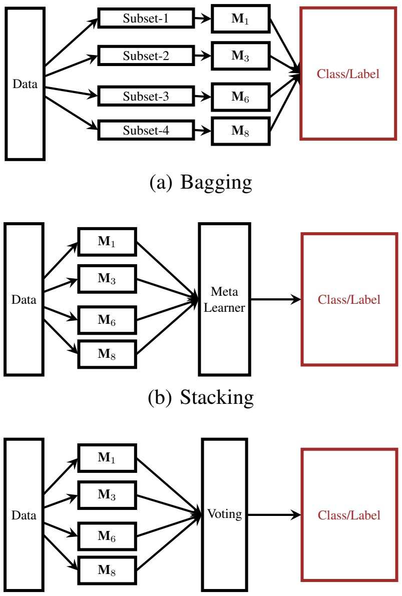
IV. Robust Strategy Against Concept Drift

To address concept drift and enhance the model's generalization, we explore an ensemble strategy. Tables I and II show that models $\mathbf{M}_1 - \mathbf{M}_8$ excel when trained and tested on identical user density. Thus, combining these models in an ensemble manner, depicted in Fig. 6, should boost generalization. We examined the subsequent ensemble methods:

- Bagging ensemble (bootstrap aggregation): It enhances the stability and accuracy of classification models. It divides the original data set $(\mathbf{X}, \mathbf{Y})$ into multiple subsets, trains the models $\mathbf{M}_1 - \mathbf{M}_8$ on each, and combines their decisions using majority voting for minimal KDR. This approach decreases variance and overfitting, shown in Fig. 6(a).

- Stacking ensemble: It trains models $\mathbf{M}_1 - \mathbf{M}_8$ separately, as in Fig. 6(b). For testing, their decisions feed a meta-learner, which integrates individual outputs for a final decision. The meta-learner uses the same LSTM-RNN architecture detailed in Section III-B3.
- Voting ensemble: It uses voting classifiers for decisions from individually trained models $\mathbf{M}_1 - \mathbf{M}_8$. The final decision picks the class with the lowest KDR, as illustrated in Fig. 6(c).

It should be highlighted that all three ensemble methods were examined using the same testing data adopted in Section III.B.



(a) Bagging

(b) Stacking

(c) Voting

Fig. 6: Illustration of the investigated ensemble strategies.

Upon training the ensemble models, test cases were considered with 1, 3, 6, and 8 users in the room. Table III compares the KDR of the three ensemble strategies, which demonstrates that the voting ensemble strategy outperforms the other strategies and offers a stable KDR of $15-16\%$ for different user densities. The KGR and NIST test results for the voting ensemble strategy are summarized in Table IV, which demonstrates that the generated keys passed all the NIST tests, and hence, are random and secure over all user densities with high KGR.

TABLE III: KDR of ensemble strategies for different test conditions.

Ensemble/Test	**1 UE**	**3 UEs**	**6 UEs**	**8 UEs**
Bagging	0.4323	0.4401	0.4478	0.4555
Stacking	0.3325	0.3502	0.3422	0.3428
Voting	**0.1495**	**0.1562**	**0.1577**	**0.1601**

TABLE IV: KGR and NIST test results of voting ensemble model for different test conditions.

Metric	**1 UE**	**3 UEs**	**6 UEs**	**8 UEs**
KGR	79.05	77.48	77.38	76
NIST Tests	Passed	Passed	Passed	Passed

V. Conclusion

This paper explores wireless secret key generation in multi-user mobile LiFi networks. Channel non-ideal reciprocity motivated an LSTM-RNN-based framework that optimizes the quantization thresholds to minimize the key disagreement rate. However, due to concept drifts in LiFi channels affected by varying user densities, the model's generalization is compromised, leading to insecure keys. A voting ensemble strategy was introduced to address this issue, ensuring robust key generation regardless of user density.

References

[1] C. Paar and J. Pelzl, *Understanding Cryptography: A Textbook for Students and Practitioners*, 1st ed. Springer, 2009.

[2] L. Bassham, A. Rukhin, J. Soto, J. Nechvatal, M. Smid, S. Leigh, M. Levenson, M. Vangel, N. Heckert, and D. Banks, "A statistical test suite for random and pseudorandom number generators for cryptographic applications," Special Publication (NIST SP), National Institute of Standards and Technology, Gaithersburg, MD, 2010, accessed June 14, 2023. [Online]. Available: https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=906762

[3] Cisco. Cisco Vision: 5G – Thriving Indoors. [Online]. Available: https://www.cisco.com/c/dam/en/us/solutions/collateral/service-provider/ultra-services-platform/5g-ran-indoor.pdf

[4] Z.-Y. Wu, M. Ismail, J. Kong, E. Serpedin, and J. Wang, "Channel characterization and realization of mobile optical wireless communications," *IEEE Trans Commun*, vol. 68, no. 10, pp. 6426–6439, 2020.

[5] C. Feng and L. Sun, "Physical layer key generation from wireless channels with non-ideal channel reciprocity: A deep learning based approach," in *2022 IEEE 95th VTC2022-Spring*, 2022.

[6] X. Zhang, G. Li, J. Zhang, A. Hu, Z. Hou, and B. Xiao, "Deep-learning-based physical-layer secret key generation for FDD systems," *IEEE Internet of Things Journal*, vol. 9, no. 8, pp. 6081–6094, 2022.

[7] M. Letafati, H. Behroozi, B. H. Khalaj, and E. A. Jorswieck, "Deep learning for hardware-impaired wireless secret key generation with man-in-the-middle attacks," in *IEEE GLOBECOM*, 2021.

[8] W. Xu, J. Zhang, S. Huang, C. Luo, and W. Li, "Key generation for internet of things: A contemporary survey," *ACM Computing Surveys*, vol. 54, no. 1, article no. 14, 2021.

[9] L. Wang, H. An, H. Zhu, and W. Liu, "Mobikey: Mobility-based secret key generation in smart home," *IEEE Internet of Things Journal*, vol. 7, no. 8, pp. 7590–7600, 2020.

[10] M. Xu, Y. Fan, and L. Liu, "Multi-party secret key generation over underwater acoustic channels," *IEEE Wireless Communications Letters*, vol. 9, no. 7, pp. 1075–1079, 2020.

[11] L. Jiao, G. Sun, J. Le, and K. Zeng, "Machine learning-assisted wireless PHY key generation with reconfigurable intelligent surfaces," in *Proceedings of the 3rd ACM Workshop on Wireless Security and Machine Learning*, ser. WiseML'21, 2021, p. 61–66.

[12] X. Wei and D. Saha, "KNEW: Key generation using neural networks from wireless channels," in *Proceedings of the 2022 ACM Workshop on Wireless Security and Machine Learning*, ser. WiseML'22, 2022, p. 45–50.

[13] E. Mahalal, M. Ismail, Z. Wu, and M. M. Fouda, "Characterization of secret key generation in 5G+ indoor mobile LiFi networks," in *GLOBECOM 2022 - 2022 IEEE Global Communications Conference*, 2022.

[14] J. Zhang, G. Li, A. Marshall, A. Hu, and L. Hanzo, "A new frontier for iot security emerging from three decades of key generation relying on wireless channels," *IEEE Access*, vol. 8, pp. 138 406–138 446, 2020.