Computer Networks and Communications
https://ojs.wiserpub.com/index.php/CNC

Research Article

Deep Learning Using Path Length Prediction for Internet of Things

Salem Omar Sati[1*], Bashir Elkharraz[2], Afif Abugharsa[2]

[1] Information Technology Faculty, Computer Networks Department, Misurata University, Misurata, Libya
[2] Information Technology Faculty, Communication and Networks Department, Misurata University, Misurata, Libya
E-mail: salem.sati@it.misuratau.edu.ly

Received: 26 December 2024; **Revised:** 14 February 2025; **Accepted:** 17 February 2025

Abstract: Sensors are employed in the Internet of Things (IoT) to collect data and establish connections with the internet. An instance of IoT can be seen in a tree-like topology constructed using wireless links. When a topology graph has a path from its root node to any other leaf or child node, and this path is influenced by the quality of wireless connections, it is known as a Destination Oriented Directed Acyclic Graph. The root node of the tree topology is responsible for implementing source routing for downstream paths to the leaf nodes of the tree. If the longest path for any node in a tested network graph, including the root, is determined by the maximum hop count, then the graph is considered to be connected. The real world and its applications are impacted by issues related to network connectivity in IoT. Models are employed to examine how changes in link probability and hop count affect the connectivity of the graph. In this research, the proposed Deep Learning (DL) model is evaluated using the Keras regression model. The simulated dataset is generated using the Cooja emulator. The link probability serves as a feature to predict the maximum hop count in the IoT. The predicted hop count based on the link probability aligns accurately with the tested data.

***Keywords*:** Internet of Things, Deep Learning, link probability, hop count, Keras model

1. Introduction

The Internet of Things (IoT) deploys sensor nodes to collect and transmit data through the internet gateway, which serves as the root of the IoT topology. IoT applications encompass various areas such as smart cities and homes, where sensors utilize wireless networks to form a tree-like topology for data collection. However, before implementing an IoT system, several challenges related to topology connection, channel and medium access, and radio coverage need to be thoroughly studied [1]. When every neighbor in the IoT's tree topology is connected through wireless links, it is referred to as a connected topology, enabling the sensors to collaborate and construct a connected graph [2]. Ensuring the transmission of sensed data to the root or internet gateway is a crucial concern for ensuring the quality of the IoT architecture. The resources of deployed sensor nodes directly impact energy consumption during connectivity in an IoT environment. While route delivery optimization and IoT topology connectivity have received attention, they still require careful consideration and extensive research [3]. Based on the aforementioned studies and papers, multi-path routing has proven to be an efficient technique for dynamic network architectures in the IoT. However, implementing multi-path routing in the IoT often demands significant processing power, leading to increased overhead and the need for further research. Recently, DL has emerged as a topic of interest in the IoT and other networking domains. DL techniques have been applied to

DOI: https://doi.org/10.37256/cnc.3120256303

routing methods as discussed in [4]. In [5], a DL algorithm was proposed, utilizing a deep Convolutional Neural Network (CNN) to address large-scale network routing and topology issues. The IoT has gained significant attention due to its broad applicability in domains such as smart cities, industrial automation, and healthcare. IoT networks often rely on wireless sensors that communicate through hierarchical structures, such as tree topologies, to optimize data transmission and network efficiency. Prior research has demonstrated the importance of structured sensor deployment in wireless environments. For instance, the paper [6] explored radix partition based over the air aggregation to improve IoT data transmission, while the paper [7] proposed a machine learning supported sensing framework for industrial IoT applications. These studies highlight the critical role of tree based wireless sensor networks in optimizing network performance, further motivating the need for predictive models to enhance IoT network reliability and scalability. In this research, we propose a model that utilizes link probability to determine the maximum hop count for IoT applications. The model is based on connection and path probabilities in relation to the hop count of a linear path. It considers the connectivity of IoT devices based on link probability and incorporates a DL framework to estimate the path hop count as an IoT routing metric. This prediction is based on connection and link connectivity, taking into account packet delivery and loss metrics. The structure of the paper is as follows: Section 2 presents an overview of the research and published articles related to IoT connectivity. Section 3 demonstrates the modeling and calculations of IoT path connectivity based on link probability. It also discusses wireless link connectivity and the formulation of path hops. Section 4 introduces the simulation model used to generate the dataset for the proposed DL model. The section also includes the radio model and connection quality used in the simulation model. Section 5 showcases the created dataset and analyzes the gathered data using distribution and correlation functions. The DL model for predicting the hop count of connected paths in the IoT is presented in Section 6. Finally, Section 7 summarizes the conclusion and suggests further areas for study.

2. Related work

Numerous research papers have examined and analyzed the topology graph from various perspectives. For instance, in one paper [8], the authors investigate the topology of an N-hop network with different packet loss probabilities in wireless links. Another paper [9] proposes a method to discover the relationship between path hop count and network topology connectivity. However, these studies mainly focus on theoretical analysis and lack predictive models for hop count estimation in real IoT deployments. Our approach fills this gap by leveraging deep learning to predict the hop count based on link probability, thus providing a practical tool for IoT topology optimization. In a different paper [10], a technique is presented for predicting the hop count in a connected topology, which is validated through simulation scenarios. Additionally, In a different paper [11], a technique is presented for predicting the hop count in a connected topology, which is validated through simulation scenarios. Additionally, another study [10] demonstrates that the probability of network connectivity is influenced by increasing radio range and discusses the impact on distribution parameters and resilience performance. In one investigation [12], two RPL objective functions are examined by arranging sky motes in different topological forms. Moreover, a mathematical model is proposed in another paper [13] that combines two important factors, namely the number of clusters and cluster size. Furthermore, a study [14] explores the relationship between message dissemination and node contact. Another paper [15] provides an introduction and comprehensive review of modern artificial intelligence developments in the context of IoT, covering architectures, techniques, and hardware platforms. Additionally, one study [16] investigates the deployment of deep learning with routing technology to improve delivery ratio when network topology changes occur. Another paper [17] proposes the distance vector hop to address node issues in specific networks. Furthermore, a paper [18] suggests a prediction routing strategy based on Markov Chains and a deep learning algorithm to predict future node locations. Finally, in a paper [19], link reliability prediction is explored to enhance the efficiency of routing protocols in IoT deployments. Unlike our work, these studies do not explore machine learning-based approaches for hop count prediction, which can enhance accuracy and adaptability in dynamic IoT environments. Our method, by utilizing a deep learning regression model, significantly improves predictive performance over traditional heuristic-based approaches. Furthermore, [20] investigates the deployment of deep learning with routing technology to improve delivery ratio when network topology changes occur. A novel routing technique based on reinforcement learning (MCTAR-SIoT) is also proposed in [21]. While these approaches integrate AI techniques into IoT routing, they primarily focus on optimizing routing paths rather than predicting hop count based on link probability. Our study introduces a data-driven prediction model that aids in understanding network connectivity trends, which can be integrated into routing decisions for enhanced efficiency. To clarify the advantages and limitations of our approach, Table 1 provides a comparative overview of existing methods and their focus areas.

Table 1. Comparison of related work and our approach

Study	Focus	Limitation	Our advantage
[8]	N-hop network topology analysis with packet loss	Lacks predictive modeling for hop count estimation	Introduces deep learning for hop count prediction
[9]	Relationship between hop count and network topology	No ML-based approach for adaptive prediction	Applies regression-based DL model for better accuracy
[10]	Hop count prediction through simulation	Lacks real-world IoT deployment validation	Uses a deep learning model trained on simulated dataset
[11]	Connectivity analysis using increasing radio range	Does not consider ML-based optimization	Incorporates link probability and hop count estimation
[12]	Evaluation of RPL objective functions on sky motes	Limited to predefined topologies	Applies DL to dynamically learn topology patterns
[13]	Mathematical model for clustering in WSN	Focuses on clustering rather than path prediction	Predicts optimal path hop count based on link probability
[14]	Message dissemination and node contact analysis	Does not consider path length prediction	Provides a predictive framework for IoT path estimation
[15]	AI review for IoT architectures and techniques	Does not propose a specific implementation	Implements a DL-based IoT topology predictor
[16]	Deep learning for routing adaptation in IoT	Focuses on routing improvements, not path estimation	Predicts hop count instead of optimizing routing directly
[17]	Distance vector hop method for node localization	Does not consider probabilistic path estimation	Uses link probability for predictive modeling
[18]	Markov Chain and DL for future node location prediction	Not designed for hop count estimation	Applies DL for accurate hop count prediction in IoT
[19]	Link reliability prediction for routing enhancement	Focuses on reliability, not hop count estimation	Combines DL with probabilistic link estimation for IoT
[20]	Deep learning for routing optimization in changing topology	Focuses on adaptive routing rather than topology prediction	Predicts hop count trends for improved network planning
[21]	Reinforcement learning-based routing (MCTAR-SIoT)	Optimizes routing but lacks topology estimation	Provides a predictive model that can enhance routing strategies
Our Work	**Deep learning-based hop count prediction**	**Predicts path length based on link probability**	**Provides a machine-learning framework for IoT topology planning**

3. Connectivity and probability of wireless link in IoT

IoT topology connectivity refers to the way sensors communicate with each other. The connectivity of the network topology is determined by how the sensors are connected to the root, which is typically connected to the internet. Routing plays a crucial role in IoT, and protocols like RPL (Routing Protocol for Low-Power and Lossy Networks) are used to discover the shortest path with the minimum number of hops. This is important for establishing an efficient end-to-end delivery between the source and the root, and the routing protocol metric, often based on hop count, determines this path. The connectivity of an IoT network is determined by the sensors' ability to find the optimal path to reach the root node. If there is no path from a sensor to the root, the data collected by that sensor cannot be processed. Path Probability ($Path_{Pro}$) is commonly used to estimate the likelihood of a path in the IoT network, and it can be calculated using the following Equation (1):

$$PP = Path_{Pro} = \sum_{i=0}^{N} LP_i \tag{1}$$

PP represents the probability of connectivity between connected sensors or sensors connected through multiple hops, such as sensor S and sensor D. LP, on the other hand, denotes the Link Probability. If two distributed sensors can communicate with each other, they are considered directly or indirectly connected, implying the existence of a single or multi-hop path between the source (S) and destination (D) sensors. Different sensor applications require a certain threshold of connectivity probability to meet their quality of service (QoS) requirements. This threshold connectivity can be achieved either by attaining a full probability of connectivity (PP = 1) or by considering the connectivity based on the number of path hops, denoted as $Path_{hops}$. The threshold of connectivity ($Conn_{th}$) is determined by the following Equation (2):

$$Conn_{th} = \begin{cases} 0 < PP \leq 1 & HC_{min} < Path_{hops} \leq HC_{max} \\ 0 & otherwise \end{cases} \tag{2}$$

HC_{min} represents the minimum hop count (HC) in the connected IoT topology, with a minimum value of two. On the other hand, HC_{max} corresponds to the maximum hop count of the path (HC), which is influenced by the Link Probability ($0 \leq LP \leq 1$). The IoT topology is established using a Destination Oriented Directed Acyclic Graph (DODAG) tree structure. The sensors involved in constructing the topology within the sensing area may experience failures, resulting in varying connectivity probabilities. These failures are often related to energy issues and can be modeled using a sensor's behavior model. The connectivity of the network is represented as a probability function of the connected path. In this paper, it is evident from the collected dataset that the number of sensors (N) and the transmission radius (r) have an impact on the network connectivity probability (PP) in terms of path hops ($Path_{hops}$). Increasing the number of sensors (N), increasing the transmission radius (r), or both, can be done to achieve a desired level of PP. The number of deployed sensors plays a vital role in enhancing the probability (PP) of network connectivity. Finally, the network connectivity probability can be simplified by assuming that the parameters r and N remain constant for a linear path. As a result, the reliable path hops ($Path_{hops}$) between the source (S) and destination (D) sensors in a connected topology can be described as following Equation (3).

$$Path_{hops} = \frac{HC_{max}}{2} Ln(LP) + HC_{max} \tag{3}$$

4. Proposed simulation model

The dataset used to analyze the connectivity of the IoT topology relies on link and path probabilities (LP, PP). To conduct this investigation, a simulation model is employed. The simulation model requires an IoT emulator capable of emulating a tree-based IoT topology that resembles real-world conditions. The emulator that fits these requirements is the Contiki/Cooja simulator [22]. The Cooja emulator not only simulates the IoT environment but also the Wireless Sensor Network (WSN) environment. It can effectively simulate both dense and sparse sensor network topologies. Contiki, a platform within the Cooja emulator, facilitates precise and detailed examination of IoT scenarios. It also enables direct emulation of sensor behavior at the physical layer. The Cooja emulator offers various radio models, including the Unit Disk Graph Medium (UDGM) with distance-loss, which is considered in this paper. Additionally, the Cooja emulator provides more realistic radio models for simulation purposes. In the case of a closed-loop IoT network where transmitters also act as receivers, the network structure may become cyclic. To accommodate this scenario, the definition of connectivity can be extended by incorporating bidirectional link probabilities, ensuring that data transmission accounts for cyclic dependencies. The proposed model can be generalized by modifying the adjacency matrix to allow for cyclic paths and adjusting the deep learning framework to learn from cyclic patterns, enabling accurate node prediction in such structures. The objective of this study is to develop a simulation methodology that creates a realistic and functional IoT network platform, with a specific focus on IoT networks. All the scenarios tested and deployed in this study as presented in Table 2 utilize the built-in Cooja radio models, particularly the Unit Disk Graph Model (UDGM). The Cooja simulator employs the UDGM radio model with distance-loss radio propagation to simulate wireless channels. This radio model assesses the quality of the communication channel based on the distance between wireless nodes. The distance is measured between directly connected nodes. The UDGM model consists of two circular disks: the transmission area and the interference area surrounding the center of the current wireless node. Within or at the interference disk, the transmitted packets from the current node can interfere with the packets sent by neighboring nodes. Successful packet delivery to the intended node occurs only when there is no interference during the transmission process. The Cooja simulator utilizes the UDGM radio model to simulate IoT networks. The UDGM considers the radio transmission range as a circular area that encompasses both interference and transmission. The range of this radio model expands with an increase in the radio output power.

Table 2. Cooja simulator settings

No	Settings	Details
1	Cooja Version	2.7
2	Radio Models	UDGM Radio Model
3	Sensor Type	Sky Mote
4	Simulation Time	5 min
5	Routing Protocol	RPL Protocol
6	Interface type	Wi-Fi 802.11
7	Performance Metric	Link Quality
8	Radio Coverage	Fixed 50 m
9	Link Quality	Based on Transmission Succeed Ratio
10	Topology	Linear Topology
11	Path Hop Count	$1 \leq HC_{max} \leq 10$

5. Implemented deep learning model

The simulation configuration and settings for different wireless IoT scenarios are presented in Table 2 . The table includes details about the network setup and parameters. It is worth noting that we have previously discussed the stability parameters of the network topology, specifically the LP and PP parameters. The analysis of path hop count ($Path_{hops}$) will be based on the LP. In the simulation, it is assumed that there is a linear path with a size of HC_{max} within the given tree topology consisting of N sensors. The collected simulation results also demonstrate the relationship between hop count and link probability in IoT, considering it as a multi-hop network. The generated data will be processed using Cooja to simulate the IoT as a network graph. The proposed DL model is implemented using the Keras framework with TensorFlow as the backend. It is designed to predict the hop count (HC) based on link probability (LP) and path probability (PP) in an IoT network. The model consists of a fully connected neural network with an input layer of two neurons representing LP and PP, followed by four hidden layers: the first with 64 neurons, the second with 128 neurons, the third with 64 neurons, and the fourth with 32 neurons, all using ReLU activation. The output layer consists of a single neuron with a linear activation function to predict the HC. The training process involves splitting the dataset into 70% for training and 30% for validation/testing, using the Adam optimizer with an initial learning rate of 0.001, which is reduced every four consecutive epochs if the validation error does not improve. The loss function used is Mean Squared Error (MSE), with a final reported validation/test MSE of 1.14×10^{-5}, indicating high prediction accuracy. The trained model is evaluated by comparing predicted and actual values using Matplotlib, demonstrating its effectiveness in accurately estimating hop count based on the simulated IoT network.

5.1 *Dataset generation and analyzing*

The path probability is determined by Equations (2) and (3), where the link quality between neighboring nodes must be sufficient to successfully transmit packets to the final destination. The hop count also affects the path probability. The dataset used in this study is generated by the Cooja simulator. It consists of features or inputs such as LP and PP, along with a label representing the HC. The dataset comprises 80 instances, each with two features (LP, PP) and one label (HC). The raw data and their relationship can be observed in Figure 1, which illustrates the relationship between LP and HC. The hop count of the linear path ranges from 1 to 10 hops, while LP varies from 0.2 to 1. Additionally, Figure 2 depicts the deployment of sky mote sensors along a linear path consisting of 10 hops. Recent research has focused on investigating the relationship between the hop count of a path and the network connectivity probability in IoT networks. This relationship is visualized in Figure 2 specifically for the UDGM radio model. It is influenced by factors such as the wireless coverage radius and the density of sensors, which directly impact the network connectivity. In order to isolate the effect of these parameters, the model assumes a circular coverage area with a fixed radius of 50 m. The model analyzes the impact of link probability on the hop count of a path, deviating from the conventional definition of network topology connectivity. Instead, it measures the connectivity by considering the link probability of the existence of a linear N-hop route in an IoT network. As the path hop count increases, the summation of link probabilities for the existence of an N-hop route decreases. This implies that two sensor nodes with a higher N-hop path distance have a lower network connectivity probability. Given a threshold link probability, the model identifies a path between source and destination sensors that has a maximum hop count of HC_{max}. To determine the value of HC_{max}, the model assumes a fixed wireless communication radius and applies the UDGM channel model.

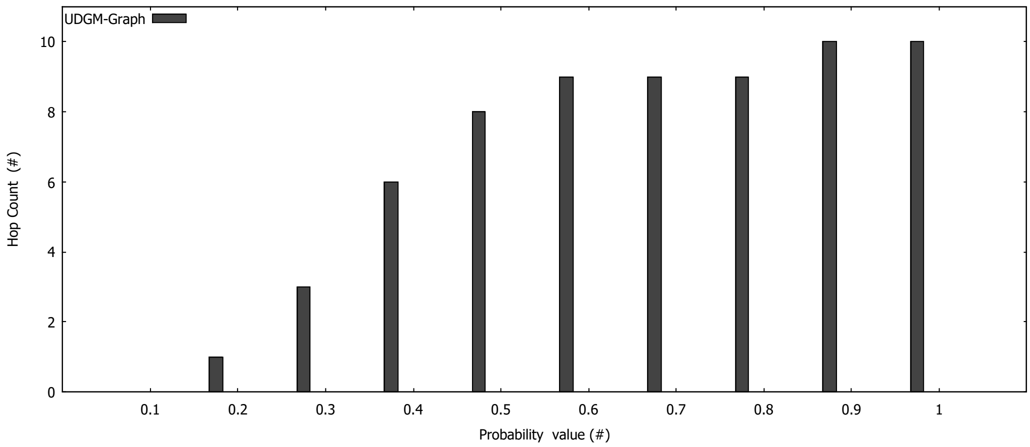

Figure 1. Link probability versus hop count of UDMG radio model

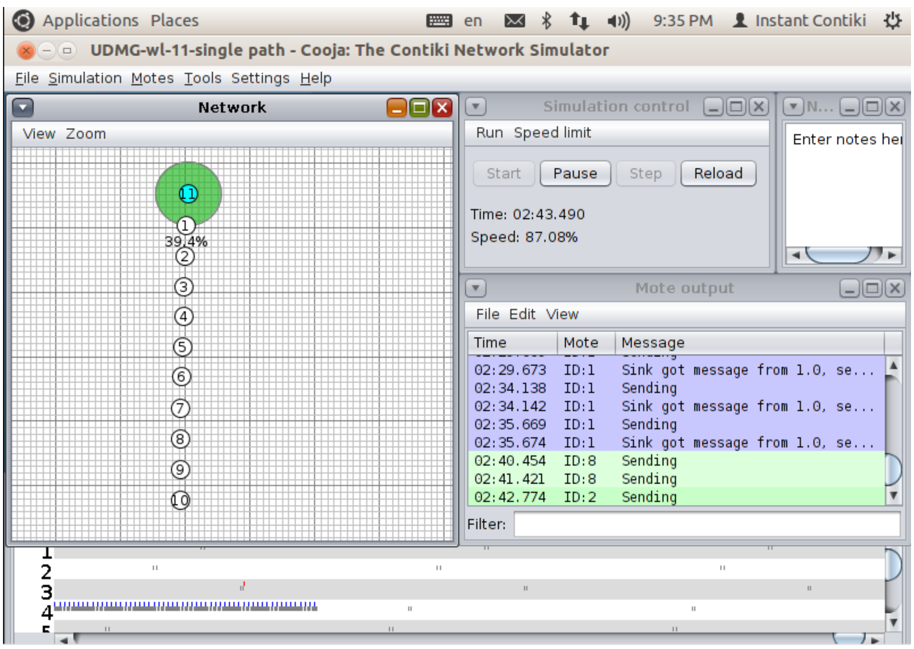

Figure 2. UDMG radio model topology for generating dataset

Furthermore, the model assumes that the IoT network is homogeneous, with a fixed radius and a defined density per unit of area. In this study, sensor nodes are considered connected if the distance between the center of their circular radio coverage is less than a specified radius, denoted as “r”. As the link probability increases, the hop count also increases until it reaches the maximum hop count, denoted as HC_{max}. The value of HC_{max} depends on the radio model used in the scenario, specifically 10 hops for the UDGM model. This relationship can be observed in Figures 1 and 2. Hence, there is a positive correlation between the maximum hop count HC_{max} and the link probability LP. When the link probability exceeds 0.7 for the UDGM radio model, the path length reaches the maximum hop count HC_{max}.

5.2 *Dataset exploring and visualization*

This section introduces the initial step of implementing a DL model for predicting the path hop count $Path_{hops}$ based on the link probability, as depicted in Figure 3. The Keras framework [23] was utilized, with Mean Squared Error (MSE) serving as the chosen loss function. The deployed Keras sequential model employed a Rectified Linear Unit (ReLU) activation function. The DL model made use of Numpy [24], Panda [25], and the plot library [24]. The simulation results were collected as a dataset from the UDGM radio model for a linear path with a maximum of ten hops, based on the link connectivity probability. This dataset was generated using the Cooja simulator, with different link probabilities as inputs (LP and PP) and the corresponding path hop counts (HC) as the outputs. The inputs were varied in steps of 0.01, ranging from 0 to 1. The dataset was loaded into the DL model as a Comma Separated Values (CSV) file, utilizing import libraries such as Panda and Numpy. The dataset consists of 80 inputs of LP and PP, along with output labels representing the number of hops ranging from 1 to 10, as illustrated in Figure 3. Additionally, the dataset file was divided into training and test/validation files, with a ratio of approximately 70% and 30% respectively. The Keras model libraries and functions were employed for this purpose. Furthermore, the dataset parameters, both features and labels, were visualized using distribution functions. For example, Figure 4 displays the distribution of LP, indicating that it varies between 0.2 and 1. The wireless channel success ratio determines the likelihood of successfully transmitting data packets across a wireless channel without errors or loss. When considering a linear or bus path with a variable path length ranging from 2 to 10 hops, the wireless channel success ratio within the range of 0.2 to 1 can be analyzed based on many factors. A channel success ratio closer to 1 indicates a higher probability of successful transmission between the source and destination nodes. Conversely, a ratio closer to 0.2 suggests a lower probability of direct transmission between the connected source and destination. The wireless channel success ratio is influenced by factors such as noise and interference. The channel success ratio between 0.2 and 1 is impacted by the path length between 2 and 10 hops in a linear path. It is important to note that in a linear path, the channel success ratio for each hop is independent of the others. Thus, the overall channel success ratio for the entire path is determined by the individual success ratios of each hop along the path. As the path length increases, the cumulative effects of link quality degradation, signal attenuation, interference, and noise can reduce the channel success ratio. Consequently, the probability of successfully transmitting a packet across the entire path may decrease with longer paths.

	LP	PP	HC
count	81.000000	81.000000	81.000000
mean	0.600000	0.162888	6.913580
std	0.235266	0.221749	3.078951
min	0.200000	0.003906	1.000000
25%	0.400000	0.013537	6.000000
50%	0.600000	0.051999	9.000000
75%	0.800000	0.240000	9.000000
max	1.000000	1.000000	10.000000

Figure 3. Dataset statistical information

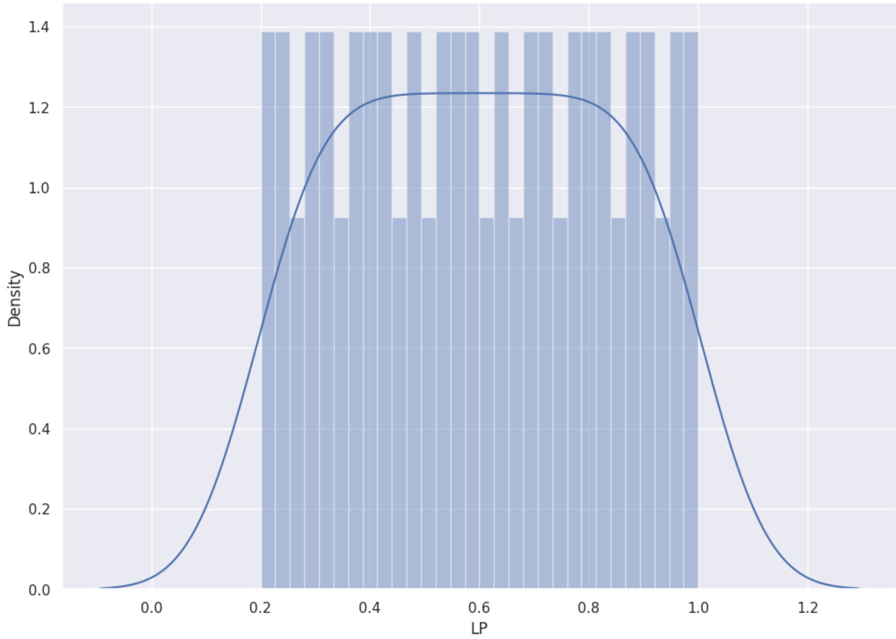

Figure 4. Link probability (LP) density of dataset

The specific channel success ratio within the range of 0.2 to 1 can vary based on factors such as signal strength, interference, noise, modulation schemes, coding techniques, and the overall quality of the wireless channel. Higher channel success ratios (close to 1) within this range signify a more reliable wireless channel and a greater likelihood of successful packet transmission. Conversely, lower channel success ratios (close to 0.2) within the specified range indicate a less reliable wireless channel, increasing the likelihood of errors or packet loss during transmission. The link probability and the semantic wireless channel are interconnected, where the characteristics of the wireless channel influence the link probability. Considering this relationship is crucial for designing and optimizing wireless networks specially to ensure reliable and efficient communication. Figure 5 shows the distribution of PP, which ranges from 0.2 to 0.4. The figure illustrates the relationship between path probability density and path length. It demonstrates that as the path length increases from 2 to 10 hops, the path probability density decreases. This phenomenon occurs because the path probability is influenced by the individual link probability or channel success ratio, which ranges from 0.2 to 1. When considering a path in a wireless network, the path probability density refers to the likelihood of successfully transmitting data packets across the entire path. It takes into account the cumulative effects of link probabilities or channel success ratios for each hop along the path. As the path length increases, the probability density decreases due to the many reasons. Such as individual link probability or channel success ratio. This factor of path probability is determined by the individual link probabilities or channel success ratios for each hop along the path. If the link probabilities or channel success ratios are lower, the overall path probability decreases. The other factor is cumulative effects where longer paths have a higher number of hops, and the cumulative effects of link degradation, signal attenuation, interference, and noise can negatively impact the link probabilities or channel success ratios. The cumulative degradation leads to a decrease in the overall path probability density.

Furthermore, the reliability which is indicates the packet transmission. the path probability density indicates a more reliable path, while a lower density suggests a less reliable path with a higher likelihood of errors or packet loss. Finally, the distribution of the HC label is demonstrated in Figure 6, revealing variations from 1 to 10 hops. The path hop count density refers to the distribution or probability density function of the number of hops required to traverse a path in a network. In this case, we are considering a liner path of 10 hops for the path length. Additionally, the success of channel transmission between 0.2 and 1 is related to this path length. The path hop count density represents the likelihood or distribution of the number of hops required to traverse a path in a network. It provides insights into the probability of encountering paths with different hop counts within a specific range. In this case, we are focusing on paths with lengths between 1 and 10 hops. The success of channel transmission refers to the probability of transmitting data across a wireless channel without errors or loss. In this context, we consider a channel transmission success probability between 0.2 and 1. A higher value indicates a higher likelihood of successful transmission, while a lower value suggests a higher chance of errors or packet loss during transmission.

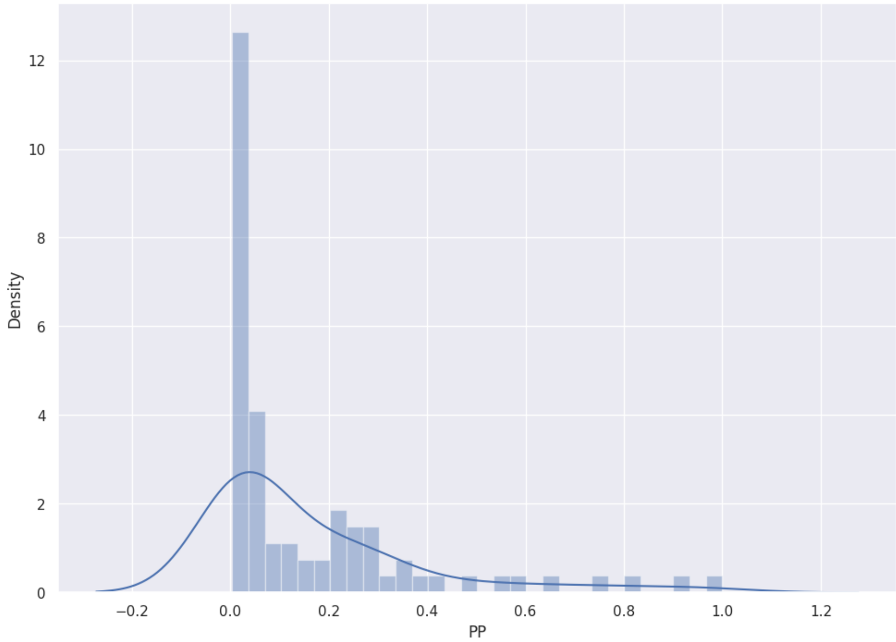

Figure 5. Path probability (PP) density of dataset

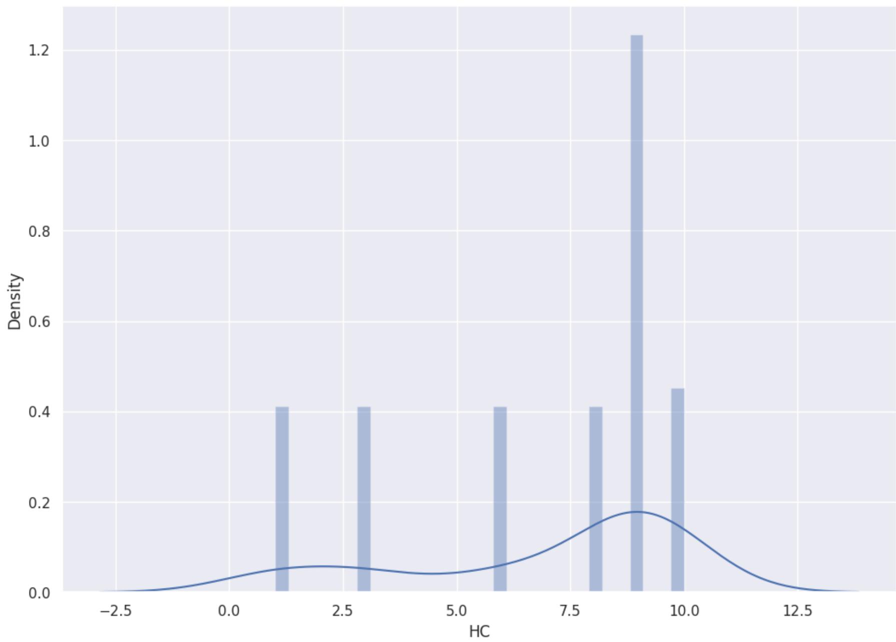

Figure 6. Hop count (HC) density of dataset

5.3 *Dataset features correlation*

The correlation between features and labels in the dataset measures the linear relationship between two features or between one feature and one label. By utilizing the correlation function provided by the seaborn library of the Keras model, it becomes possible to predict a desired label based on the test features. The rationale behind using the correlation function for feature selection in the DL model is to identify the most relevant variables as the best features. These selected features should exhibit a high correlation with the target or desired label, while being uncorrelated with each other as input to the model. If two features are correlated, the DL model can predict a label using only one of the highly correlated features. Therefore, the correlation function is employed to ensure that no two features are highly correlated with each other. The DL model only requires one highly correlated feature, as the other highly correlated features do not provide additional information to the model. In this case, Pearson correlation is used. A threshold value is set, such as 0.5, as an absolute value for selecting the variables. If multiple features are correlated, the model should discard the feature with a lower correlation value compared to the target or desired label. Based on Figure 7, which illustrates the correlation function between features LP and PP, along with the label HC, there are some correlations observed. For instance, LP and PP have a correlation coefficient of 0.53, while both LP and PP correlate with the label HC at coefficients of 0.91 and 0.22 respectively. Therefore, the DL model utilizes LP and PP as features, with HC as the desired output label.

Figure 7. Features and labels correlation

6. Hop count prediction using deep learning

The DL model is implemented using the Keras framework, consisting of input and output layers, as well as hidden layers. Specifically, a Convolutional Neural Network (CNN) with four dense hidden layers is constructed. This sequential model is compiled with Mean Squared Error (MSE) as the loss function and ReLU as the activation function. Numpy, seaborn, and Panda are utilized for visualizing and exploring the dataset, while the plot library is used for plotting correlations and density. Additionally, the plot library is employed to visualize the predicted values of HC based on the tested values of LP, as shown in Figure 8. The training process is conducted using Keras as the backend API of Tensorflow [26]. The Adam optimizer is chosen to enhance the prediction performance. The datasets are transformed into data frames using Pandas and Numpy. The data is split into training data (70%) for features and labels, and the remaining 30% is used for validation or testing purposes. The learning rate is carefully selected and reduced by a factor of two every four consecutive epochs if the validation error does not decrease significantly. The reported mean squared error (MSE) on the validation or test data is 1.14×10^{-5}. After training the model, future predictions are made and evaluated. The trained Keras model is used to pass the training, testing, and validation data through the CNN in order to obtain predictions for the desired outputs. The Matplotlib library is employed to generate a plot illustrating the test and predicted values for the hop count (HC), as depicted in Figure 8. This plot demonstrates that the constructed sequential CNN Keras model successfully predicts the path hop count values compared to the simulated graph shown in Figure 1.

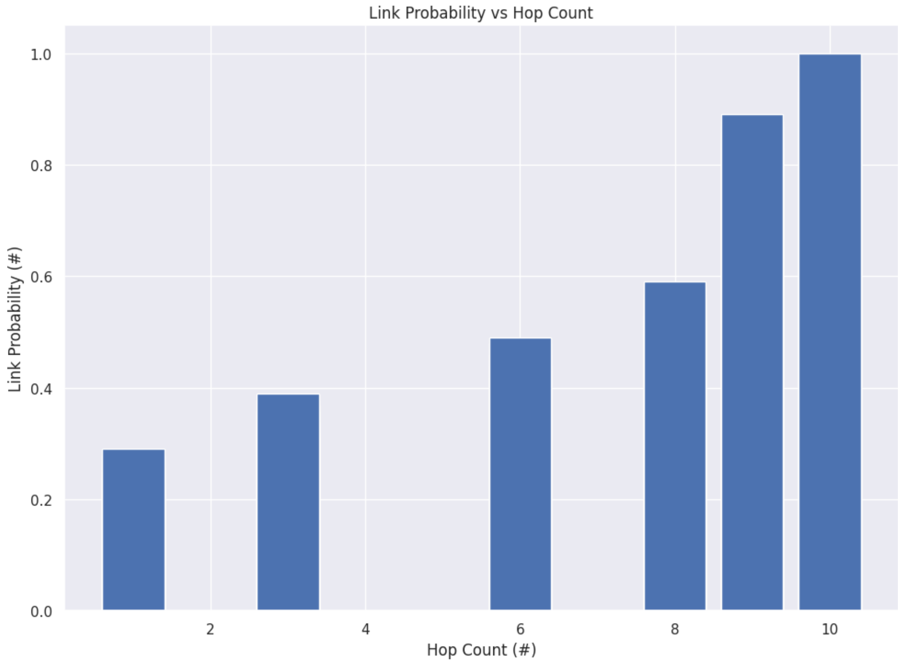

Figure 8. Hop count prediction using deep learning

7. Conclusions and future work

This paper introduces a regression deep learning (DL) model built using the Keras framework to explore the impact of network connectivity on the Internet of Things (IoT). The raw data is obtained by simulating the UDGM radio model using the Cooja simulator. The investigation focuses on path hops, utilizing link probability and path probability as features to predict the linear path of the IoT and determine the maximum hop count. The main contribution of this work lies in the integration of deep learning techniques, specifically a Convolutional Neural Network (CNN), to model network connectivity in IoT environments. This provides a novel approach for optimizing path predictions and improving the efficiency of network design.

The proposed DL model, a CNN deep learning model, is implemented using the Keras framework as a nonlinear regression model. The evaluation of the model demonstrates a close alignment between the predicted values and the trained values, with a mean squared error (MSE) of 1.14×10^{-5}. This performance suggests the potential of deep learning for network connectivity prediction in IoT, with implications for enhancing routing protocols and optimizing resource allocation.

In future research, it is suggested to expand the radio models to incorporate more realistic Cooja radio models, such as the MRM model. This expansion would provide a more accurate representation of real-world IoT environments, where wireless propagation may vary due to factors like interference, mobility, and environmental conditions. Additionally, further studies could focus on extending the model to include a broader range of IoT scenarios, including diverse node densities, varying transmission ranges, and more complex mobility patterns. This would allow for a deeper understanding of the dynamic nature of network connectivity and its impact on IoT performance.

Confilict of interest

There is no conflict of interest for this study.

References

[1] S. Nematzadeh, M. Torkamanian-Afshar, A. Seyyedabbasi, and F. Kiani, "Maximizing coverage and maintaining connectivity in WSN and decentralized iot: an efficient metaheuristic-based method for environment-aware node deployment," *Neural Comput. Appl.*, vol. 35, no. 1, pp. 611–641, 2023.

[2] Y. Li, G. Jiang, M. Wu, and Y. Song, "Methods to measure the network path connectivity," *Secur. Commun. Netw.*, vol. 2020, pp. 8836590:1–8836590:10, 2020.

[3] G. Ijemaru, K. L. Ang, and J. K. Seng, "Wireless power transfer and energy harvesting in distributed sensor networks: Survey, opportunities, and challenges," *Int. J. Distrib. Sens. Netw.*, vol. 18, no. 3, p. 155014772110677, 2022.

[4] B. Mao, Z. M. Fadlullah, F. Tang, N. Kato, O. Akashi, T. Inoue, et al., "Routing or computing? the paradigm shift towards intelligent computer network packet transmission based on deep learning," *IEEE Trans. Comput.*, vol. 66, no. 11, pp. 1946–1960, 2017.

[5] Z. Md. Fadlullah, F. Tang, B. Mao, J. Liu, and N. Kato, "On intelligent traffic control for large-scale heterogeneous networks: A value matrix-based deep learning approach," *IEEE Commun. Lett.*, vol. 22, no. 12, pp. 2479–2482, 2018.

[6] M. Tang, S. Cai, and V. K. N. Lau, "Radix-partition-based over-the-air aggregation and low-complexity state estimation for iot systems over wireless fading channels," *IEEE Trans. Signal Process.*, vol. 70, pp. 1464–1477, 2022.

[7] Z. Guo, Y. Shen, C. Chakraborty, F. Alblehai, and K. Yu, "Industrial 6g-iot and machine-learning-supported intelligent sensing framework for indicator control strategy in sewage treatment process," *IEEE Internet Things J.*, vol. 11, no. 18, pp. 29308–29320, 2024.

[8] O. Ayan, H. M. Gursu, A. Papa, and W. Kellerer, "Probability analysis of age of information in multi-hop networks," *IEEE Netw. Lett.*, vol. 2, no. 2, pp. 76–80, 2020.

[9] W. Liao, M. Wu, M. Zhao, P. Li, and T. Li, "Hop count limitation analysis in wireless multi-hop networks," *Int. J. Distrib. Sens. Netw.*, vol. 13, no. 1, 2017.

[10] Y. Huang, J. Martínez, J. Sendra, and L. López-Santidrián, "The influence of communication range on connectivity for resilient wireless sensor networks using a probabilistic approach," *Int. J. Distrib. Sens. Netw.*, vol. 9, 2013.

[11] O. D. Jerew and K. L. Blackmore, "Estimation of hop count in multi-hop wireless sensor networks with arbitrary node density," *Int. J. Wirel. Mob. Comput.*, vol. 7, no. 3, pp. 207–216, 2014.

[12] F. M. Alatersh, S. Sati, and M. Sullabi, "Impact of network topologies on RPL performance," in *Proc. 22nd Int. Arab Conf. Inf. Technol. (ACIT 2021)*, Muscat, Oman, Dec. 21–23, 2021, pp. 1–7.

[13] A. Sohoub, , S. Sati, and M. Eshtawie, "Optimal cluster size for wireless sensor networks," *Int. J. Wirel. Microw. Technol.*, vol. 13, no. 1, pp. 36–44, Feb. 2023.

[14] A. Ippisch, S. Sati, and K. Graffi, "Device to device communication in mobile delay tolerant networks," in *Proc. 21st IEEE/ACM Int. Symp. Distrib. Simul. Real Time Appl. (DS-RT 2017)*, Rome, Italy, Oct. 18–20, 2017, A. D'Ambrogio, R. E. D. Grande, A. Garro, and A. Tundis, Eds. Computer Society, 2017, pp. 91–98.

[15] K. P. Seng, L. Ang, and E. Ngharamike, "Artificial intelligence internet of things: A new paradigm of distributed sensor networks," *Int. J. Distrib. Sens. Netw.*, vol. 18, no. 3, p. 155014772110628, 2022.

[16] A. Zhang, M. Sun, J. Wang, Z. Li, Y. Cheng, and C. Wang, "Deep reinforcement learning-based multi-hop state-aware routing strategy for wireless sensor networks," *Appl. Sci.*, vol. 11, no. 10, 2021.

[17] "A novel distance vector hop localization method for wireless sensor networks," *J. Intell. Syst.*, vol. 32, no. 1, Jan. 2023.

[18] G. A. Montoya, C. Lozano-Garzon, and Y. Donoso, "Energy-efficient and delay sensitive routing paths using mobility prediction in mobile wsn: Mathematical optimization, markov chains, and deep learning approaches," *IEEE Access*, vol. 9, pp. 153382–153400, 2021.

[19] K. V. Bhaskar, T. Kumanan, S. Sree Southry, and V. Elangovan, "Deep neural network algorithm to improve link reliability in wireless sensor networks," in *Proc. 2023 4th Int. Conf. Electron. Sustain. Commun. Syst. (ICESC 2023)*, Coimbatore, India, Jul. 6–8, 2023, pp. 476–480.

[20] A. Hameed, J. Violos, and A. Leivadeas, "A deep learning approach for iot traffic multi-classification in a smart-city scenario," *IEEE Access*, vol. 10, pp. 21193–21210, 2022.

[21] A. Kaur, R. Sahu, and D. Anand, "Deep reinforcement base multi context routing in social iot," in *Proc. 2023 Int. Conf. Circ. Power Comput. Technol. (ICCPCT 2023)*, Kollam, India, Aug. 10–11, 2023, pp. 675–682.

[22] T. Mehmood, "COOJA network simulator: Exploring the infinite possible ways to compute the performance metrics of IOT based smart devices to understand the working of IOT based compression & routing protocols," *CoRR*, vol. abs/1712.08303, 2017.

[23] D. Diouf, A. Niang, and S. Thiria, "Deep learning based multiple regression to predict total column water vapor (TCWV) from physical parameters in west africa by using keras library," *CoRR*, vol. abs/1912.07388, 2019.

[24] L. Roubeyrie and S. Celles, "Windrose: A python matplotlib, numpy library to manage wind and pollution data, draw windrose," *J. Open Source Softw.*, vol. 3, no. 29, p. 268, 2018. https://doi.org/10.21105/joss.00268.

[25] W. McKinney, "Data structures for statistical computing in python," in *Proc. 9th Python Sci. Conf. (SciPy 2010)*, Austin, TX, USA, Jun. 28–Jul. 3, 2010, S. van der Walt and J. Millman, Eds., 2010, pp. 56–61.

[26] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, et al., "Tensorflow: Large-scale machine learning on heterogeneous distributed systems," *CoRR*, vol. abs/1603.04467, 2016.