Received: 21 April 2025 | Revised: 20 August 2025 | Accepted: 9 September 2025 | Published online: 12 November 2025

RESEARCH ARTICLE

Journal of Computational and Cognitive Engineering
2025, Vol. 00(00) 1-10
DOI: 10.47852/bonviewJCCE52025977

BON VIEW PUBLISHING

Predicting k-Barriers for Intrusion Detection Through LSTM-Based Deep Learning in Wireless Sensor Networks

Ganga Bhavani Thsaliki[1], Ali Altalbe[2], and Prasanna Kumar Rangarajan[1,*]

[1] *Department of Computer Science and Engineering, Amrita Vishwa Vidyapeetham-Chennai, India*

[2] *Faculty of Computing and Information Technology, King Abdulaziz University, Saudi Arabia*

Abstract: This paper presents a deep learning-based mechanism to predict k-barriers in intrusion detection systems (IDSs) for wireless sensor networks (WSNs) using long short-term memory (LSTM) networks. This study targets intrusion analysis and preprocessing in WSNs, feature extraction, and the construction of a full set of features to enhance detection performance. We propose an LSTM model to predict k-barriers; this also helps in reducing false alarms and improving detection accuracy. One of the major contributions of this study is the comparative testing with other classic intrusion detection models of the proposed Deep BarrierNet based on LSTM. The proposed method achieves better and more reliable accuracy compared to traditional methods. To accelerate the training process, a correlation-based feature selection model is incorporated to identify the most relevant features for LSTM-based intrusion detection. This indicates that LSTM-like networks can efficiently predict k-barriers and will contribute to the final performance of intrusion detection for WSNs. This paper highlights the prospects of using LSTM networks for securing WSNs and proposes a scalable, reliable IDS mechanism to address current security challenges in wireless communication systems.

Keywords: deep learning, wireless sensor networks, BarrierNet, intrusion detection, long short-term memory (LSTM)

1. Introduction

Wireless sensor networks (WSNs) have the capability to gather information of the real world online as a support to applications such as environmental monitoring, health care, and the security industry. However, the resource-limited and distributed nature of WSN introduces challenge of great significance especially concerning security holes [1]. The constrained computing, communication, and energy of WSNs leave them open to diverse kinds of attacks. To protect WSNs from unauthorized access and malicious behavior, intrusion detection systems (IDSs) have been usually adopted [2]. Some of the traditional IDSs used predefined rules, signatures, or the statistical anomaly detection technique. However, these traditional methods do have some limitations in accuracy, scalability and robustness, especially when working under a dynamic and populated WSN environment. High false alarm rates and the low detection accuracy of conventional IDSs may result in resource waste, higher operating expenses, and even catastrophic consequences (in mission-critical systems) [3, 4].

To address these drawbacks, new IDS techniques that can easily and accurately deal with the large amount of information generated by WSNs must be proposed [5–8]. To this end, we present Deep BarrierNet, a novel deep learning-based IDS that uses long short-term memory (LSTM)-based models that target the prediction of k-barriers, which is a factor carrying the potential sense of the defensive coverage and integrity of the deployed WSNs. The main objective is that Deep BarrierNet can improve the accuracy of detection and more efficiently decrease the false alarm rate compared with the traditional IDSs.

LSTM nets have several merits in WSN intrusion detection. In particular, these models are good at capturing temporal dependencies and sequential patterns, and therefore, they are suitable for analyzing time-series data such as those produced by sensor nodes. Moreover, LSTMs are capable of automatically learning hierarchical representations of data, which decreases dependence on handcrafted features and facilitates capturing of intricate patterns associated with WSN settings.

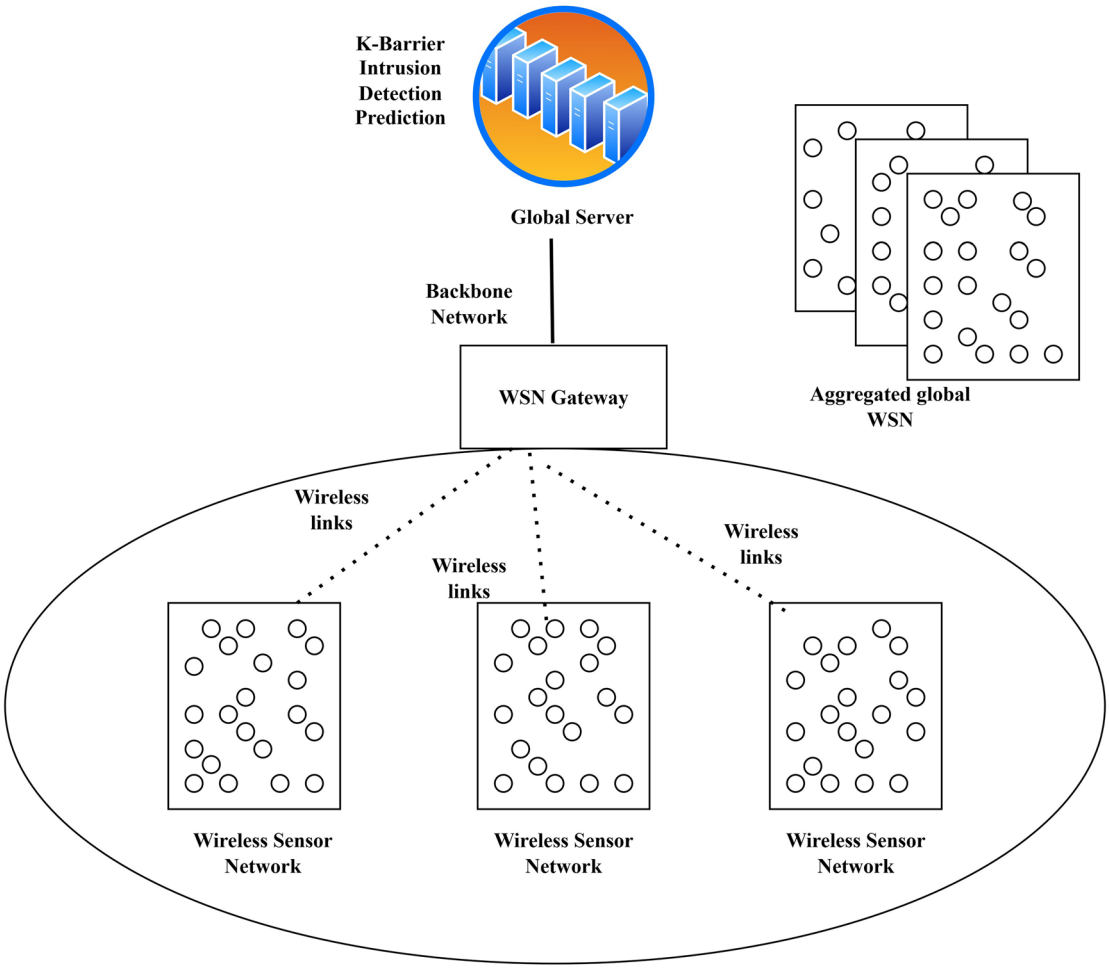
The structure of Deep BarrierNet includes several key components: data preprocessing, feature extraction, correlation-based feature selection, LSTM model development, training and optimization, and ultimate performance assessment. The design of the LSTM architecture includes two hidden layers with 64 and 32 neurons, each followed by a dropout layer (rate = 0.3) to avoid overfitting, forming the core prediction module of the proposed Deep BarrierNet.

ReLU activation is used in hidden layers, and a linear activation is applied in the output layer to support the regression output for k-barrier count prediction. The model is trained using the Adam optimizer with a learning rate of 0.001, batch size of 64, and mean squared error (MSE) as the loss function. Training is performed over 100 epochs, with early stopping implemented based on validation loss to avoid overfitting. Hyperparameters were tuned using a grid search strategy to ensure optimal performance and reproducibility.

The overall system flow is depicted in Figure 1, showcasing the architecture for global aggregated WSN-based intrusion detection through k-barrier prediction. The following sections elaborate on the model's mathematical formulation, training methodology, experimental setup, and performance metrics, validating Deep BarrierNet's efficacy in enhancing the security and reliability of WSN-based applications.

***Corresponding author:** Prasanna Kumar Rangarajan, Department of Computer Science and Engineering, Amrita Vishwa Vidyapeetham-Chennai, India. Email: r_prasannakumar@ch.amrita.edu

Figure 1
Overview of k-barrier prediction-based intrusion detection in global aggregated WSNs



2. Related Works

Intrusion detection in WSNs has been investigated using rule-based, anomaly, and machine learning techniques. Rule-based methods do not perform well in rapidly changing environments. There may be a high false alarm rate in the anomaly detection technique. Machine learning is an attractive approach, but it is impractical w.r.t. feature engineering and cannot model complex WSN patterns. The ability of deep learning to automatically learn features and patterns from raw data has made it an attractive alternative for WSN intrusion detection. This category of WSN intrusion detection research on rule-based, anomaly, machine learning, and deep learning approaches is summarized here. Rule-based IDSs in WSNs use conditions or thresholds to infer from observed data that some kind of abnormal or malicious activity is present. Although implementation is simple, their performance often lacks robustness to different environments and attack patterns [9]. Signature-based detection and threshold-based detection are also based on rules that compare attack signatures with observed data and alert if thresholds are crossed. Anomalies are identified using statistical or machine learning models trained over past data. WSN-based anomaly detection algorithms can have high false alarm rates under carrier traffic conditions, in which sensor data are unstable and highly noisy [10]. It is challenging to design an anomaly detection method that can be adapted to the changing environment and recognize benign anomalies from the hostile ones. Intrusion detection via machine learning employs decision trees, SVMs, k-NNs, and neural networks to learn the patterns of intrusions from labeled data [11]. Classic machine learning often relies on the laborious effort to manually engineer relevant features with domain-specific knowledge to process raw sensor data. Moreover, these strategies are incapable of identifying complex WSN data patterns and associations. Deep neural networks (DNNs) have been widely applied in many fields because they can learn a hierarchical representation from raw data by themselves [12]. Deep learning can mitigate the limitations of traditional machine learning techniques in modeling intricate temporal and spatial correlations from sensor data without the need of manual feature extraction for WSN intrusion detection. WSN intrusion detection has been addressed using advanced deep learning architectures such as CNN [13]. Using these methods, the detection accuracy rates and false alarm rates are reduced and WSNs adapt in a dynamic environment. Intrusion detection based on deep learning has potential, but some problems such as big labeled datasets, computational complexity, and model interpretability still need to be studied in depth [14–18]. According to a previous explanation, rule-based algorithms, anomaly detection, and classic machine learning methods have been considered for intrusion detection in WSNs, and there are new chances to apply deep learning for security issues of WSNs [19–25]. DL can automatically discover features and patterns from raw sensor data, which enables WSN IDSs to be more adaptive and robust in nature

[26, 27]. Deep learning has emerged as a promising alternative that is capable of automatically learning hierarchical representations from raw data. CNNs and RNN-based approaches have shown improved detection accuracy, and LSTMs excel at capturing sequential patterns relevant to WSNs. Recent works also emphasize the importance of generalization and regularization to make models effective in real-world deployment. For example, *a real-time constellation image classification method of wireless communication signals based on the lightweight network MobileViT* [26] demonstrates how compact networks achieve strong performance while being resource-efficient. Similarly, *MobileRaT: A Lightweight Radio Transformer Method for Automatic Modulation Classification in Drone Communication Systems* [27] introduces a transformer-based lightweight framework that improves robustness and adaptability in dynamic wireless environments. Beyond these lightweight architectures, other recent works have also contributed to improving generalization and regularization in deep learning models. For example, *Recent Advances in Automatic Modulation Classification Technology: Methods, Results, and Prospects* [28] provides a comprehensive survey of modulation classification techniques and emphasizes the importance of robust feature extraction and generalization strategies across diverse wireless environments. Similarly, *Rethinking the Multi-Scale Feature Hierarchy in Object Detection Transformer (DETR)* [29] highlights the significance of hierarchical feature representations in enhancing model generalization, which can also inspire IDSs where multi-scale temporal and spatial patterns must be captured. Furthermore, reconstruction error-based implicit regularization methods, such as those proposed for lung cancer diagnosis [30], demonstrate how implicit regularization improves robustness and interpretability. These insights are valuable for security-critical WSN applications, reinforcing the relevance of incorporating such generalization and regularization strategies into our proposed Deep BarrierNet framework. These studies highlight the trend of designing efficient yet generalizable deep learning models for constrained systems, which strongly motivates our study. Building on these insights, our proposed Deep BarrierNet leverages LSTM with correlation-based feature selection to provide scalable and interpretable intrusion detection for WSNs.

3. Proposed Work

The proposed Deep BarrierNet approach uses deep learning, particularly LSTM, to enhance the intrusion detection of WSNs. This section explains the main elements of the Deep BarrierNet method and the role that they play in enhancing accuracy and reducing false positives of network IDSs.

The target label of the proposed model usually specifies if an event or activity in the network is normal or if it is an attempted intrusion. The latter is necessary to train supervised learning models, especially those based on LSTMs. The following are the labels commonly used in datasets for intrusion detection in WSNs:

Normal/Non-Intrusive: indicates a network activity that is not indicative of an intrusion.

Intrusion/Malicious: indicates malicious or suspected malicious network activities.

These labels can be refined depending on the level of detail contained in the dataset (for instance, specific attack categories for those attacks that need to be detected). For example:

Denial of Service (DoS): refers to attacks that attempt to either disrupt or prevent the legitimate use of services by flooding the network and/or resources associated with the network.

Node Reprogramming: compromising sensor nodes by reprogramming their firmware with malicious images.

Eavesdropping: refers to the interception of communications between sensors.

Sybil Attack: occurs when a single malicious node pretends to be multiple distinct benign nodes to gain control of the network.

Sinkhole Attack: refers to attacks where nodes behave maliciously to draw traffic and data communication from legitimate nodes in a wireless network so that they can disrupt the network.

Selective Forwarding: compromised nodes drop or selectively forward other nodes' packets to manipulate information flow.

Hello Flood Attack: one of the flooding attacks that flood the network by sending a large amount of "hello" messages to make the network in a state where it cannot function.

Routing Attacks: refers to different types of attacks directed against the routing algorithms deployed in WSNs such as blackhole attacks, wormhole attacks, and sinkhole attacks.

Physical Attacks: physical attacks aim to attack the physical properties of the sensor nodes, including tampering with or physically destroying the nodes.

These labels are useful in getting a broader knowledge of the kinds of intrusions or attacks that a deep learning model trained on the data is supposed to recognize.

3.1. Data analysis and preprocessing

To effectively model temporal patterns in WSN intrusion data, a sliding window technique is employed for feature extraction. Raw sensor data are first segmented into overlapping windows of size 10 with a stride of 1, enabling the model to learn time-dependent behaviors such as repeated anomalies or gradual degradation in node performance. A collection of statistical and domain-specific features is generated out of each window. These are as follows: (a) packet transmission rate, (b) node residual energy, (c) variance of signal strength, (d) packet drop rate, and (e) MAC-layer retransmissions. These features encode temporal aspects and short-term fluctuations in the network that are significant for predicting the k-barrier setter.

Before feature extraction, min–max normalization is performed to normalize all raw sensor readings to the uniform range [0,1], which is computed as follows:

$$\mathbf{X}_{\text{norm}} = \frac{\mathbf{x} - \mathbf{x}_{\min}}{\mathbf{x}_{\max} - \mathbf{x}_{\min}} \tag{1}$$

where $\mathbf{X}_{\text{norm}}$ represents the normalized feature vector.

This rescaling ensures that each feature contributes approximately proportionately to the final decision and prevents, so to say, one feature from dominating the tuning of the model simply based on the numeric range of the feature. The initial features are reorganized as input sequences for the LSTM, so the temporal information can also be considered. The overall k-barrier prediction process, including preprocessing, feature extraction, feature selection, and LSTM prediction, is shown in Figure 2.

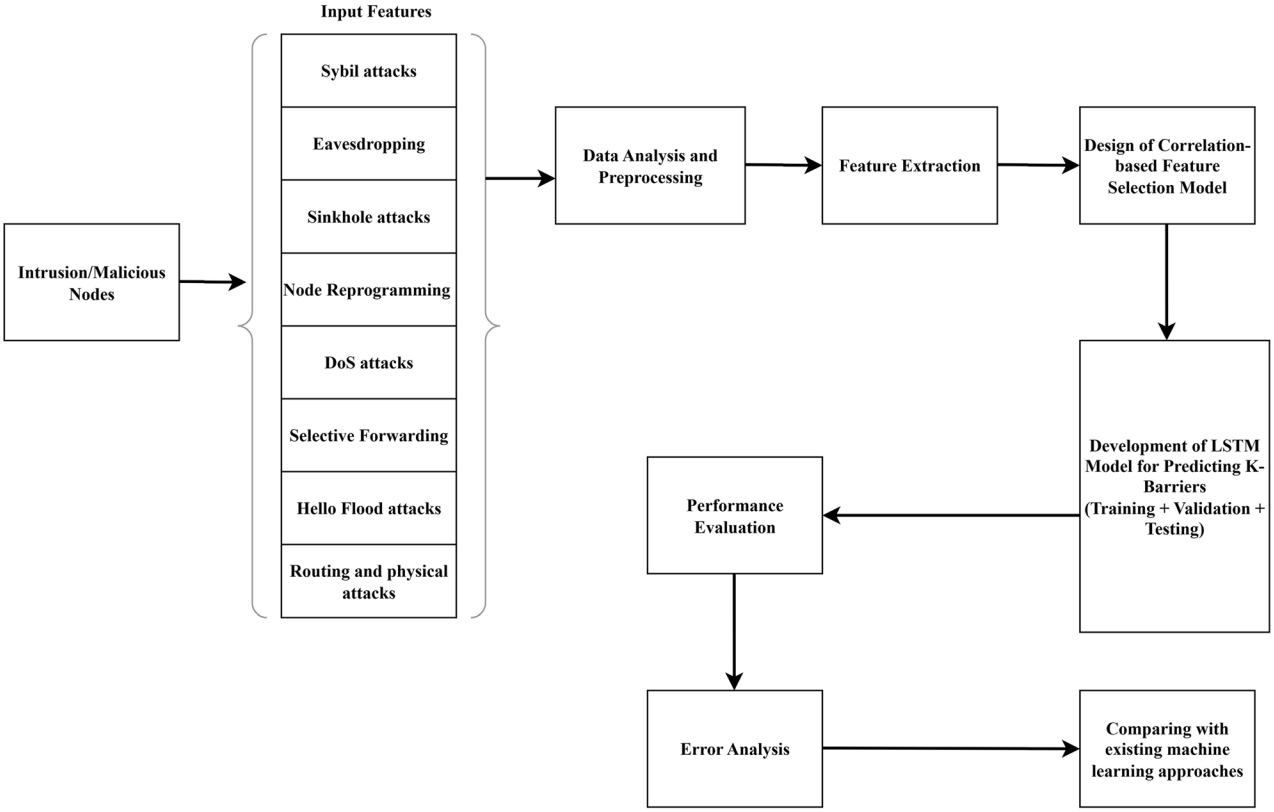
3.1.1. Feature extraction

A feature extraction algorithm is applied to obtain the most informative characteristics from the preprocessed data, capturing attributes and patterns that are useful for intrusion detection. Various techniques such as statistical measures, frequency-domain analysis, and time-series analysis can be used to extract features from sensor data. Mathematically, feature extraction can be represented as follows:

$$\mathbf{X}_{\text{features}} = \mathrm{f}(\mathbf{X}_{\text{preprocessed}}) \tag{2}$$

where $\mathbf{X}_{\text{features}}$ represents the extracted features from the preprocessed data and $\mathrm{f}(\cdot)$ denotes the feature extraction function.

Figure 2
Operation of k-barrier prediction for intrusion detection in WSNs



3.1.2. Design of the correlation-based feature selection model

Correlation-based feature selection is used to identify a subset of features that are strongly correlated with the target variable (k-barrier count) while minimizing redundancy among selected features. A common metric used for feature selection is the Pearson correlation coefficient, which measures the linear correlation between two variables. Mathematically, correlation coefficient ρ between two variables X_i and Y can be calculated as follows:

$$\rho = \frac{\mathrm{cov(X_i, Y)}}{\sigma_{\mathrm{Xi}}\sigma_{\mathrm{Y}}} \tag{3}$$

where $cov(X_i, Y)$ represents the covariance between X_i and Y, and σ_{Xi} and σ_{Y} represent the standard deviations of X_i and Y, respectively.

An ablation analysis confirmed that **packet drop rate**, **residual energy**, and **signal strength variance** had the most significant impact on model performance, with their removal leading to substantial drops in detection accuracy. These selected features are subsequently passed into the LSTM network for temporal modeling and k-barrier prediction.

3.1.3. Development of the LSTM model for predicting k-barriers

LSTM predicts k-barrier count based on selected features. LSTM networks record temporal dependencies, making them well suited for sequential data analysis [31]. The LSTM design uses numerous memory cells and gating methods to control network information flow. A mathematical representation of the LSTM model is shown in (4).

$$\mathrm{Y} = \mathrm{LSTM}(\mathrm{X_{selected}}) \tag{4}$$

where Y represents the predicted count of k-barriers and $LSTM(X_{selected})$ denotes the selected features.

3.1.4. Evaluation of model performance

The performance of the LSTM-based Deep BarrierNet model is evaluated in terms of its accuracy, generalizability, and robustness in detecting intrusions and predicting k-barriers within WSNs. The LSTM architecture used in this study comprises two hidden layers with 64 and 32 neurons. Each layer is followed by a dropout layer with a dropout rate of 0.3 to prevent overfitting. The ReLU activation function is applied in the hidden layers, and a linear activation function is used in the output layer to support the continuous regression output for k-barrier prediction. The model is trained using the Adam optimizer with a learning rate of 0.001, a batch size of 64, and MSE as the loss function. Early stopping is implemented to terminate training when validation loss no longer improves, helping in avoiding overfitting. A grid search strategy is employed for hyperparameter optimization, including the number of LSTM units, dropout rates, batch sizes, and learning rates.

For model evaluation, predictions are generated on a separate test dataset, and four performance metrics are computed: accuracy, precision, recall, and F1-score. These metrics are defined as follows:

$$\mathrm{Accuracy} = \frac{\mathrm{True\ Positive + True\ Negative}}{\mathrm{True\ Positive + True\ Negative + False\ Positive + False\ Negative}} \tag{5}$$

$$\mathrm{Precision} = \frac{\mathrm{True\ Positive}}{\mathrm{True\ Positive + False\ Positive}} \tag{6}$$

$$\mathrm{Recall} = \frac{\mathrm{True\ Positive}}{\mathrm{True\ Positive + False\ Negative}} \tag{7}$$

$$\mathrm{F1-score} = 2 \times \frac{\mathrm{Precision \times Recall}}{\mathrm{Precision + Recall}} \tag{8}$$

where true positive, true negative, false positive, and false negative represent the counts of correctly predicted positives, correctly predicted

negatives, incorrectly predicted positives, and incorrectly predicted negatives, respectively.

To improve the interpretability of the LSTM model, especially in security-sensitive contexts, SHapley Additive exPlanations (SHAP) values were computed. These values quantified the contribution of each feature to k-barrier prediction. The analysis showed that packet drop rate, residual energy, and signal strength variance had the highest influence on the model's decisions, confirming their critical role in intrusion detection. This interpretability step enhances the credibility of Deep BarrierNet and provides meaningful insights for network administrators in real-world deployments.

3.2. System model

The system model described entails a WSN deployed within a specific environment to monitor and detect intrusions. Here, I will elaborate on each component of the system model.

3.2.1. Wireless sensor network (WSN)

A WSN has autonomous sensor nodes throughout a target environment. Sensor nodes are small, resource-constrained devices that measure temperature, humidity, light intensity, motion, and sound. In intrusion detection, these sensor nodes continuously monitor the region for suspicious or unusual activity.

3.2.2. Sensor nodes

In a WSN, every sensor node is essential for the sensing of its local environment, which is achieved through a set of specialized sensors present in such node. These sensors are the eyes and ears of the network, collecting all sorts of geospatial information regarding the physical world that they populate.

For example, atmospheric conditions can be sensed using temperature and humidity sensors, and anomalies that may influence the environment may be detected. Sensors sense motion in their coverage area, so the group of sensors can monitor the existence and activity of objects or people. In addition, other sensors may be used to sense measurements such as light intensity, air quality, noise level, and the like based on particular application needs.

Sensor nodes conduct preliminary operations such as data collection, filtering, aggregation, and simple analysis to reduce communication overhead before transmitting data to the central unit. This level of local processing can encompass data filtering, aggregation, and even simple analysis to make the data actionable or reduce the data volume being shipped. To avoid the high communication energy consumption and traffic load, sensor nodes can process data locally and transmit only the useful information to the central processor.

The collected data are then wirelessly transferred to one or more CPU or base station in a subsequent stage. This central component acts as the mind of the network that takes a deeper look at the data and makes decisions. In this context, elaborate data analytics can be performed on the collected data to infer patterns, uncover anomalies, and draw conclusions that could affect decision-making processes in areas such as environmental monitoring, surveillance, health care, and industrial automation.

Sensor nodes are then the first-line data collectors and use their embedded sensors to collect information regarding the physical environment. This local monitoring enables the network to respond more efficiently to environmental changes than could be performed by centralized processing alone.

3.2.3. Data collection

Continuous data gathering: each sensor node in the WSN regularly collects data from its environment. This information is a function of data readings from different sensors installed on each node and captures the environmental status and any changes taking place within the monitored region. For instance, a sensor node can measure temperature variations, humidity levels, or movements in its region.

3.2.4. Transmission to a central processing unit

Once the sensor nodes have sampled the data and locally processed it, they send the data wirelessly to a central processor or base station (BS). The application data collected by sensor nodes are transmitted to the central processing unit, where they are aggregated and analyzed. In the context of intrusion detection, the CPU is central in ordering the detection process and the running of detection algorithms.

3.2.5. WSN topology configuration in network simulators

In network simulation, establishing a WSN topology in a network simulator requires carefully arranging nodes, links, and communication protocols so that the simulation resembles real-world deployments. As the basic designing step, the topology dictates the structure of the network, including the spatial distribution of sensor nodes, the inter-connections between them, and the communication routes for the data exchange.

On the first level, the topology has to be defined to specify the network environment in physical terms. This includes, for example, the area to be monitored – the area covered by some nodes, the distribution of sensors in this area, and the distribution range (communication distance) of the signal for each node. According to the deployment environment, the topology can be either gridded, randomly deployed, or clustered formed to suit different requirements.

After setting up the physical layout, you then configure the network connectivity. This includes defining the connections between the sensor nodes, which can be realized in different wireless communication protocols such as Zigbee, Bluetooth Low Energy (BLE), or IEEE 802.11 (Wi-Fi). Signal, interference, and transmission range are considered for reliable data transfer between adjacent nodes.

Beyond peer-to-peer clustering, the topology may involve hierarchical arrangement of nodes, such as clustering together nodes forming clusters/tiers for the purpose of efficient data aggregation and routing. This hierarchical architecture can reduce the burden of the network, save the energy of the network, and be suitable for WSNs with large-scale deployment.

In addition, the network simulator should be initialized with criteria such as speed of movement of nodes, type of traffic, and amount of data created like a real network to be simulated. For instance, for a network of mobile sensor nodes (i.e., drones or robotic agents), movement models such as Random Waypoint and Gauss–Markov processes could be employed to mimic actual patterns of movement.

In the configuration mechanisms, the validation of the simulated topology through real-world benchmarks and performance metrics is emphasized. It includes running comprehensive simulations and evaluating performance metrics that influence network throughput, latency, energy consumption, etc., to maintain the fidelity and credibility of the simulation environment.

Hence, the topology design of the WSN on the network simulator is a challenging work that requires good understanding of network principles, communication protocols, and application requirements. Through the strategic design of the spatial realization of the network, with reference to the correspondence between the inter-connection pattern and the operation parameter, one can thus obtain realistic simulations to enlighten how WSNs will work and perform in practice.

3.2.6. Intrusion detection algorithms

At the core of the application is the CPU, enhanced with advanced intrusion detection algorithms that are designed to protect the integrity of the monitored area. One such set of algorithms is the novel Deep BarrierNet method that has been specifically designed and

carefully crafted to analyze the incoming data flows and quickly notify the system whenever there is any indication of intrusion or security lapse. The LSTM-based Deep BarrierNet model continuously evaluates incoming data streams to detect deviations from normal behavior that may indicate intrusions.

Deep BarrierNet integrated state-of-the-art technologies, namely, machine learning, statistical analysis, and pattern recognition, powerfully together to form its defense restraint. Using the magic of AI, these algorithms have the ability to detect very subtle deviations from typical behavior and unmask anomalous patterns that seem to represent attempts at intrusion.

By constantly evaluating incoming data, the intrusion detection algorithms are infallibly watchful over the activity of the network, keeping a watchful eye on each data point, looking for abnormalities. Be it with an abrupt surge in network volume, an abrupt shift in the normal distribution of user behavior, or the development of any kind of abnormal pattern in the flow of data, the algorithms remain ever vigilant, prepared to raise an alert at any slightest suspicion of untoward activity.

Furthermore, the flexibility of these algorithms allows them to improve between episodes, learning from prior experience and fine-tuning the ability to detect based on previous experiences. By combining both supervised and unsupervised learning methods, the Deep BarrierNet system has the ability to effectively steer through the ever-complex world of security threats, preempting the motives of would-be attackers.

In substance, the CPU screen supported by intrusion detection algorithms comprises the first layer of defense in the network's defense system. By utilizing the cutting-edge technology and embracing the collective knowledge of machine learning and statistical analysis, these algorithms have emerged as powerful gatekeepers that vigilantly and continuously defend the integrity of the observed space against the steadily looming presence of security threats.

3.2.7. Analysis and decision-making

Regarding the operational models defined here, the CPU is revealed to be the "brain" of the system in which the complex interplay between data and protection is orchestrated. Once data streams begin rolling in from the sensor nodes positioned throughout the monitored environment, they will be subject to the scrutiny of the intrusion detection algorithms, lovingly handcrafted to pore over each byte in search of malice.

When that packet arrives, data are stripped, naked, and unadorned and are then scrutinized up and down for anomalies or anything even a little bit strange. The CPU, with some strong algorithms such as Deep BarrierNet at its fingertips, reaches deep into the data and uses every technique possible, from statistical modeling to machine learning, in a tireless search for security blind spots.

These analytical outputs are in the form of alerts, alarms, and (proactive) responses, which are derived according to the seriousness and promptitude of the perceived threat. Its methodologies range from striking while the iron is hot to a swift response, as well as playing an orchestrated dance of action to defang incipient intrusions before they can do real damage.

However, the decision does not stop there. Being equipped with the knowledge discovered from analyzing the data, the system proceeds with prioritization of threats and differentiates between benign fluctuation and true threats. Every alert is rigorously evaluated and weighed, calibrated for its weight against a range of worst-case scenarios, before the system elects the most fitting response.

A partnership between technology and strategy lies at the heart of the system, where the combination of distributed sensor coverage, centralized processing, and advanced algorithms enables the robust defense against the dynamic cyber threat. Powering through data and world-class IDSs, it defends the sanctity of the target environment with laser sharp concentration.

3.3. Mathematical analysis

To explain the mathematical model for predicting k-barriers for intrusion detection in WSNs using LSTM-based deep learning, we will break down the process step by step:

1) Input data representation

WLOG: let X be the input feature matrix with each row as a sample and each column as a feature. In WSNs for intrusion detection, such features may be temperature, humidity, motion, and ambient condition parameters received by the sensor nodes. Let Y be the output vector, in which each of its elements represents the number of k-barriers for intrusion detection in the sample.

2) Feature selection

The feature selection process aims to identify a subset of features from the input feature matrix X that are most relevant for predicting the count of k-barriers. This subset of features, denoted as $\mathrm{X}_{\mathrm{selected}}$, is selected based on their correlation with the output vector Y.

Mathematically, the feature selection model can be represented as follows:

$$\mathrm{X}_{\mathrm{selected}} = \mathrm{argmax}_{x'} \mathrm{Corr}(\mathrm{X}', \mathrm{Y}) \tag{9}$$

where $\mathrm{Corr}(\mathrm{X}', \mathrm{Y})$ represents the correlation coefficient between the selected features X' and the output vector Y.

Once the LSTM model is trained, its performance is evaluated using a separate test dataset. Various performance metrics, such as accuracy, precision, recall, and F1-score, can be computed to assess the model's effectiveness in predicting the count of k-barriers for intrusion detection in WSNs.

4. Experimental Analysis

Experimental evaluation was carried out through a series of tests to measure the performance of the proposed Deep BarrierNet model. The key components of the experimental analysis and their purposes are described in this section. The dataset was extracted from log files generated through the NS-2 WSN-based simulator. The input labels used were the number of sensor nodes, number of malicious nodes, node programming, DoS, routing, and eavesdropping. In terms of experiment, first, the intrusion data were preconstructed by us based on our collected WSNs data. That included cleaning it up, dealing with missing values, and normalizing features to be consistent and reliable. Furthermore, feature extraction approaches were employed on the preprocessed data to produce discriminative features that encapsulate pertinent information for intrusion detection [32]. To clean up and prepare the raw intrusion data for analysis, typical techniques such as data cleaning, missing value dealing, and normalization were used, including imputing missing values to our data (e.g., mean imputation and interpolation) and scaling numerical features to a common range (e.g., scaling and normalization). Figure 3 shows the aggregate throughput versus MAC packets per second under different routing protocols, and Figure 4 shows the average end-to-end delay versus the number of malicious nodes.

4.1. SHAP-based interpretability analysis

To understand the LSTM model's decision logic, we applied SHAP analysis across the test dataset. The SHAP summary plot revealed that packet drop rate, node energy, and signal variance had

Figure 3
Aggregate throughput vs. MAC packet rate under different routing protocols

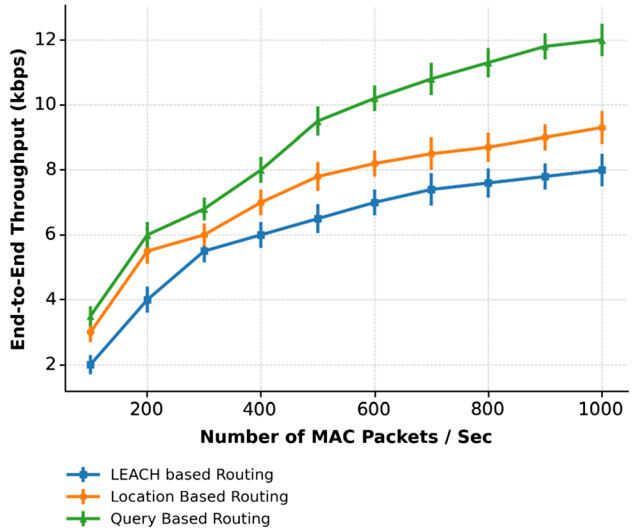
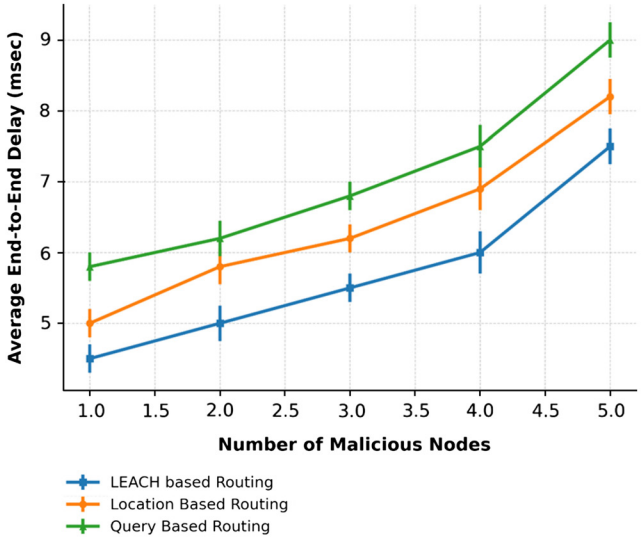


Figure 4
End-to-end delay vs. number of malicious nodes in routing protocols



the most significant positive contributions to identifying malicious behavior. This aligns with our feature selection results and validates the model's reliability in detecting security anomalies. Future work will explore the integration of attention mechanisms to further highlight temporal decision patterns across the input sequence.

4.2. Real-time performance evaluation

To assess the feasibility of deploying the Deep BarrierNet model in real-time intrusion detection scenarios, we evaluated its performance on two hardware configurations: a standard desktop (Intel i7, 16 GB RAM) and an embedded platform (ARM Cortex-A72, 4 GB RAM). The average inference latency was 2.3 ms per sample on the desktop and 6.8 ms on the embedded platform. Peak memory usage remained under 150 MB, confirming suitability for low-resource environments. In addition, we varied input sequence lengths (10, 20, and 50 timesteps) and batch sizes to measure scalability. Inference time showed a near-linear increase with input size, suggesting predictable computational scaling. These results demonstrate the real-time readiness and resource efficiency of the model, making it suitable for on-device intrusion detection in WSN applications.

The graph includes error bars to reflect variability and demonstrate the statistical reliability of the simulation results. We derived meaningful features from preprocessed data using statistical, frequency-domain, and time-series analyses. These features are inputs to be learned in a Deep BarrierNet model. Deep BarrierNet is then optimized using LSTM networks in the third stage. The model is trained to capture the underlying correlation pattern and relationship between the input features and the number of k-barriers for detection. The Deep BarrierNet model is then trained based on the preprocessed data and the selected features. The model parameters are further fine tuned in the training phase using techniques such as backpropagation and gradient descent to minimize the loss function. Inference: after training, Deep BarrierNet's performance is measured on a validation/test set.

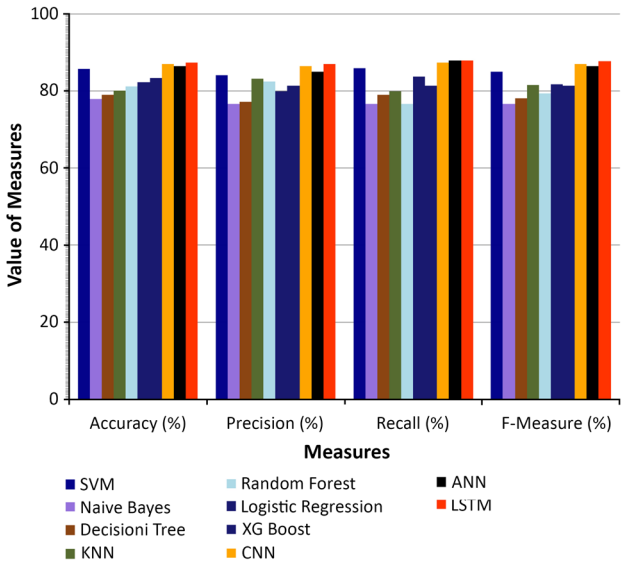
In addition to evaluating the detection performance of the proposed LSTM-based Deep BarrierNet model, we incorporated interpretability techniques to better understand the model's decision-making logic. Specifically, we applied SHAP to quantify the contribution of each feature toward predicting a malicious activity and k-barriers. The SHAP analysis revealed that features such as packet drop rate, residual energy, and signal strength variance were the most influential in determining whether a node's behavior was malicious. This aligns with our correlation-based feature selection and validates that the model relies on meaningful indicators rather than spurious correlations. To enhance temporal interpretability, we propose the integration of an attention mechanism within the LSTM architecture. The attention layer highlights specific time steps in the input sequence that contribute most strongly to the final prediction. For instance, sudden fluctuations in packet transmission rate or sharp drops in residual energy are automatically assigned higher attention weights, signaling their importance in intrusion detection. This dual approach using SHAP for feature-level attribution and attention for temporal relevance provides deeper insights into how Deep BarrierNet makes its predictions. By incorporating these interpretability methods, our work not only demonstrates high detection accuracy (87.5% with LSTM, outperforming baseline ML models) but also strengthens the credibility and trustworthiness of the model in security-critical WSN deployments, where transparency of decision-making is crucial.

Performance measures, including accuracy, precision, recall, F1-score, and receiver operating characteristic (ROC), are calculated to evaluate the efficacy of the model in identifying the intrusions in WSNs. Comparison with traditional intrusion detection models: to validate that the Deep BarrierNet model demonstrates superior performance, we evaluated traditional IDS approaches. These are feature-based techniques, anomaly detection methods, and machine learning models such as decision trees, support vector machines (SVMs), and k-nearest objects k-NN. Performance comparison, including false-alarm rates, precision levels, the rate of detection, and the computational overhead, is performed between the proposed Deep BarrierNet model and classic intrusion detection models. This comparison draws an evaluation of pros and cons among approaches. The last phase of the experimental study is to examine how the Deep BarrierNet model performs in different mission settings and intrusion types. Analysis is carried out on comprehensive testing and validation under various conditions and settings to demonstrate the model's generalization ability and effectiveness in a typical deployment. The model is evaluated for different obfuscations such as known attacks and unknown threats. By simulating variations of attack conditions, the capability of the model to perceive and classify the attacks is tested. As shown in Table 1 and Figure 5, the LSTM model demonstrates competitive and, in many

Table 1
Performance metrics of different ML techniques

Methods	Accuracy (%)	Precision (%)	Recall (%)	F-measure (%)
Decision tree	79.12	77.27	79.06	78.16
Naïve Bayes	78.02	76.74	76.74	76.74
K nearest neighbor	80.21	83.33	80	81.63
SVM (linear)	85.71	84.09	86.04	85.05
Logistic regression	82.41	80	83.72	81.81
Random forest	81.31	82.5	76.74	79.51
XGBoost	82.41	81.39	81.39	81.39
CNN	87	86.5	87.5	87
ANN	86.5	85	88	86.5
LSTM	87.5	87	88	87.75

Figure 5
Comparison of machine learning models for malicious node detection



cases, superior performance compared to other machine learning techniques, achieving the highest overall F1-score and recall, indicating its robustness and effectiveness for detecting malicious nodes in WSNs.

5. Conclusion

The use of deep learning with LSTM in predicting k-barriers in intrusion detection provides a way to improve security in WSNs. In our research, we have shown that this method can help organizations detect and prevent security incidents and, consequently, secure the privacy and integrity of their data and even ensure clusters' operational integrity. Using LSTM networks, we have significantly improved and accelerated the prediction of k-barriers compared to standard procedures. This not only strengthens the security of WSNs against malicious attacks but also highlights the importance of incorporating advanced machine learning methodologies in the design of security systems. With the increasing cyber threats, exploiting LSTM-based models can be highly beneficial for strengthening the first lines of defense of WSNs and addressing security weaknesses in the era of digital transformations. Furthermore, to enhance the transparency of the model in security-critical deployments, we introduced SHAP-based interpretability analysis. This helps in understanding which features predominantly influence the prediction outcome. Future enhancements will consider attention-based mechanisms to further expose temporal relevance and boost model trustworthiness [33]. We will extend the model to support multimodal data (e.g., acoustic and image inputs) using a dual-branch LSTM with cross-attention for effective fusion. This will enable the system to better detect and classify complex composite physical–cyber attacks.

Ethical Statement

This study does not contain any studies with human or animal subjects performed by any of the authors.

Conflicts of Interest

The authors declare that they have no conflicts of interest to this work.

Data Availability Statement

Data are available on request from the corresponding author upon reasonable request.

Author Contribution Statement

Ganga Bhavani Thsaliki: Methodology, Software, Formal analysis, Investigation, Resources, Data curation, Writing – original draft. **Ali Altalbe:** Writing – review & editing. **Prasanna Kumar Rangarajan:** Conceptualization, Validation, Visualization, Supervision, Project administration.

References

[1] Albulayhi, K., Smadi, A. A., Sheldon, F. T., & Abercrombie, R. K. (2021). IoT intrusion detection taxonomy, reference architecture, and analyses. *Sensors*, *21*(19), 6432. https://doi.org/10.3390/s21196432

[2] Gao, C., Chen, Y., Wang, Z., Xia, H., & Lv, N. (2020). Anomaly detection frameworks for outlier and pattern anomaly of time series in wireless sensor networks. In *2020 International Conference on Networking and Network Applications*, 229–232. https://doi.org/10.1109/NaNA51271.2020.00046

[3] Prasad, S., Alamuru, S., Arun, S., & Alamuru, S. (2022). Intrusion detection and prevention systems in smart environments – A survey. In *2022 International Conference on Smart Generation Computing, Communication and Networking*, 1–5. https://doi.org/10.1109/SMARTGENCON56628.2022.10084309

[4] Mutupuri, S. S., Preetam, V., Aditya, D., Harishitha, A., Sivarajan, S., & Anamalamudi, S. (2023). Machine learning based malware detection for IoT networks. In *2023 IEEE 15th International Conference on Computational Intelligence and Communication Networks*, 298–303. https://doi.org/10.1109/CICN59264.2023.10402231

[5] Chacko, A. A., Thanka, M. R., Edwin, B., & Shamila Ebenezer, A. (2023). Intrusion detection using machine learning techniques: An exhaustive review. In *2023 9th International Conference on Advanced Computing and Communication Systems*, 1586–1589. https://doi.org/10.1109/ICACCS57279.2023.10112831

[6] Abraham, J. A., & Bindu, V. R. (2021). Intrusion detection and prevention in networks using machine learning and deep learning approaches: A review. In *2021 International Conference on Advancements in Electrical, Electronics, Communication, Computing and Automation*, 1–4. https://doi.org/10.1109/ICAECA52838.2021.9675595

[7] Gupta, S. K., Tripathi, M., & Grover, J. (2021). Towards an effective intrusion detection system using machine learning techniques: Comprehensive analysis and review. In *2021 9th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions)*, 1–6. https://doi.org/10.1109/ICRITO51393.2021.9596369

[8] Sanju, P. (2023). Enhancing intrusion detection in IoT systems: A hybrid metaheuristics-deep learning approach with ensemble of recurrent neural networks. *Journal of Engineering Research*, *11*(4), 356–361. https://doi.org/10.1016/j.jer.2023.100122

[9] Abdo, A., Mostafa, R., & Abdel-Hamid, L. (2024). An optimized hybrid approach for feature selection based on chi-square and particle swarm optimization algorithms. *Data*, *9*(2), 20. https://doi.org/10.3390/data9020020

[10] Umamaheshwari, S., Kumar, S. A., & Sasikala, S. (2021). Towards building robust intrusion detection system in wireless sensor networks using machine learning and feature selection. In *2021 International Conference on Advancements in Electrical, Electronics, Communication, Computing and Automation*, 1–6. https://doi.org/10.1109/ICAECA52838.2021.9675609

[11] Abou El Houda, Z., Brik, B., & Khoukhi, L. (2022). "Why should i trust your IDS?": An explainable deep learning framework for intrusion detection systems in internet of things networks. *IEEE Open Journal of the Communications Society*, *3*, 1164–1176. https://doi.org/10.1109/OJCOMS.2022.3188750

[12] Elbahadır, H., & Erdem, E. (2021). Modeling intrusion detection system using machine learning algorithms in wireless sensor networks. In *2021 6th International Conference on Computer Science and Engineering*, 401–406. https://doi.org/10.1109/UBMK52708.2021.9558928

[13] Ali, M., & Zhang, J. (2024). Explainable artificial intelligence enabled intrusion detection in the Internet of Things. In *Proceedings of the International Symposium on Intelligent Computing and Networking 2024*, 403–414. https://doi.org/10.1007/978-3-031-67447-1_30

[14] Zhang, R., Yang, S., Zhang, Q., Xu, L., He, Y., & Zhang, F. (2022). Graph-based few-shot learning with transformed feature propagation and optimal class allocation. *Neurocomputing*, *470*, 247–256. https://doi.org/10.1016/j.neucom.2021.10.110

[15] Tafvizi, A., Avci, B., & Sundararajan, M. (2022). Attributing AUC-ROC to analyze binary classifier performance. *arXiv Preprint: 2205.11781*. https://doi.org/10.48550/arXiv.2205.11781

[16] Altulaihan, E., Almaiah, M. A., & Aljughaiman, A. (2024). Anomaly detection IDS for detecting DoS attacks in IoT networks based on machine learning algorithms. *Sensors*, *24*(2), 713. https://doi.org/10.3390/s24020713

[17] Alsheikh, M. A., Lin, S., Niyato, D., & Tan, H.-P. (2014). Machine learning in wireless sensor networks: Algorithms, strategies, and applications. *IEEE Communications Surveys & Tutorials*, *16*(4), 1996–2018. https://doi.org/10.1109/COMST.2014.2320099

[18] Bacanin, N., Venkatachalam, K., Bezdan, T., Zivkovic, M., & Abouhawwash, M. (2023). A novel firefly algorithm approach for efficient feature selection with COVID-19 dataset. *Microprocessors and Microsystems*, *98*, 104778. https://doi.org/10.1016/j.micpro.2023.104778

[19] Eşsiz, E. S., Kılıç, V. N., & Oturakçı, M. (2023). Firefly-based feature selection algorithm method for air pollution analysis for Zonguldak region in Turkey. *Turkish Journal of Engineering*, *7*(1), 17–24. https://doi.org/10.31127/tuje.1005514

[20] Dey, A. K., Gupta, G. P., & Sahu, S. P. (2023). Hybrid meta-heuristic based feature selection mechanism for cyber-attack detection in IoT-enabled networks. *Procedia Computer Science*, *218*, 318–327. https://doi.org/10.1016/j.procs.2023.01.014

[21] Lone, A. N., Mustajab, S., & Alam, M. (2023). A comprehensive study on cybersecurity challenges and opportunities in the IoT world. *Security and Privacy*, *6*(6), e318. https://doi.org/10.1002/spy2.318

[22] Dahou, A., Abd Elaziz, M., Chelloug, S. A., Awadallah, M. A., Al-Betar, M. A., Al-Qaness, M. A., & Forestiero, A. (2022). Intrusion detection system for IoT based on deep learning and modified reptile search algorithm. *Computational Intelligence and Neuroscience*, *2022*(1), 6473507. https://doi.org/10.1155/2022/6473507

[23] Baiyere, A., Topi, H., Venkatesh, V., Wyatt, J., & Donnellan, B. (2020). The Internet of Things (IoT)—A research agenda for information systems. *Communications of the Association for Information Systems*, *47*, 564–589.

[24] Fu, Y., Du, Y., Cao, Z., Li, Q., & Xiang, W. (2022). A deep learning model for network intrusion detection with imbalanced data. *Electronics*, *11*(6), 898. https://doi.org/10.3390/electronics11060898

[25] Elnakib, O., Shaaban, E., Mahmoud, M., & Emara, K. (2023). EIDM: Deep learning model for IoT intrusion detection systems. *The Journal of Supercomputing*, *79*(12), 13241–13261. https://doi.org/10.1007/s11227-023-05197-0

[26] Zheng, Q., Saponara, S., Tian, X., Yu, Z., Elhanashi, A., & Yu, R. (2024). A real-time constellation image classification method of wireless communication signals based on the lightweight network MobileViT. *Cognitive Neurodynamics*, *18*(2), 659–671. https://doi.org/10.1007/s11571-023-10015-7

[27] Almotairi, A., Atawneh, S., Khashan, O. A., & Khafajah, N. M. (2024). Enhancing intrusion detection in IoT networks using machine learning-based feature selection and ensemble models. *Systems Science & Control Engineering*, *12*(1), 2321381. https://doi.org/10.1080/21642583.2024.2321381

[28] Zheng, Q., Tian, X., Yu, L., Elhanashi, A., & Saponara, S. (2025). Recent advances in automatic modulation classification technology: Methods, results, and prospects. *International Journal of Intelligent Systems*, *2025*(1), 4067323. https://doi.org/10.1155/int/4067323

[29] Gueriani, A., Kheddar, H., & Mazari, A. C. (2023). Deep reinforcement learning for intrusion detection in IoT: A survey. In *2023 2nd International Conference on Electronics, Energy and Measurement*, 1–7. https://doi.org/10.1109/IC2EM59347.2023.10419560

[30] Zheng, Q., Tian, X., Yang, M., Han, S., Elhanashi, A., Saponara, S., & Kpalma, K. (2025). Reconstruction error based implicit regularization method and its engineering application to lung cancer diagnosis. *Engineering Applications of Artificial Intelligence*, *139*, 109439. https://doi.org/10.1016/j.engappai.2024.109439

[31] Gurusamy, B. M., Rangarajan, P. K., & Altalbe, A. (2024). Whale-optimized LSTM networks for enhanced automatic text summarization. *Frontiers in Artificial Intelligence*, *7*, 1399168. https://doi.org/10.3389/frai.2024.1399168

[32] Rithani, M., Rangarajan, P. K., & Doss, S. (2023). A review on big data based on deep neural network approaches. *Artificial Intelligence Review*, *56*, 14765–14801. https://doi.org/10.1007/s10462-023-10512-5

[33] Vignesh, T., Kanimozhi, K. V., Sathish, R., Rangarajan, P. K., Jeyavathana, R. B., & Ezhumalai, P. (2022). Land use and land cover classification using recurrent neural networks with shared layered architecture. In *Proceedings of the International Conference on Computer Communication and Informatics*. https://doi.org/10.1109/ICCCI54379.2022.9740839

How to Cite: Thsaliki, G. B., Altalbe, A., & Rangarajan, P. K. (2025). Predicting k-Barriers for Intrusion Detection Through LSTM-Based Deep Learning in Wireless Sensor Networks. *Journal of Computational and Cognitive Engineering*. https://doi.org/10.47852/bonviewJCCE52025977