International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

# Detecting Security Threats in Wireless Sensor Networks using Hybrid Network of CNNs and Long Short-Term Memory

**Gopala T.[1], Raviram V.[2], Udaya Kumar N. L.[3]**

**Submitted**: 24/06/2023 **Revised**: 08/08/2023 **Accepted**: 27/08/2023

**Abstract**: Wireless Sensor Networks (WSNs) are pervasive in various domains due to their capability to monitor and collect data from the environment. However, the open and distributed nature of WSNs makes them susceptible to security threats and attacks. Detecting and mitigating these attacks is vitalto confirming the veracity and reliability of the collected data. In this study, we propose a novel CNN-LSTM hybrid network that makes use of the geographical and temporal information found in sensor data to identify attacks in WSNs. The proposed hybrid network combines the advantages of long short-term memory (LSTM) networks and convolutional neural networks (CNNs). CNNs are used to automatically extract key features from the sensor input and learn spatial representations. In order to identify the temporal dependencies and long-term patterns present in the consecutive sensor readings, the output of the final CNN layer is then fed into the LSTM layers. We conducted experiments utilizing a real-world WSN dataset encompassing a variety of typical actions and various types of attacks to assess the efficacy of our technique. The dataset was carefully curated and labeled to ensure its representativeness and diversity. Our CNN-LSTM hybrid network was evaluated against a number of baseline models, such as standalone CNN and LSTM networks and conventional machine learning techniques frequently employed for attack detection in WSNs. The experimental findings show that, in terms of accuracy, precision, recall, and F1-score, our proposed CNN-LSTM hybrid network outperforms the baseline models. The hybrid network's ability to capture both spatial and temporal information allows it to better discern subtle attack patterns that might be missed by the standalone CNN or LSTM models. Furthermore, the model exhibits robustness and generalization, effectively detectingvarious attack scenarios while maintaining low false positive rates.

***Keywords****: Wireless Sensor Network, Deep Learning,LSTM, CNN, Security Issues.*

## 1. Introduction

In a wide range of industries, including environmental monitoring, healthcare, industrial automation, and smart cities, Wireless Sensor Networks (WSNs) have become a game-changing innovation. These networks are made up of numerous inexpensive sensor nodes which work together to cooperatively collect and send data from the physical environment to a central base station. The collected data is used for real-time monitoring, decision-making, and analysis, leading to improvements in efficiency, resource management, and overall quality of services.

However, the open and resource-constrained nature of WSNs exposes them to various security threats and vulnerabilities. As WSNs operate in unattended and potentially hos- tile environments, they become targets for malicious attacks, including unauthorized access, data tampering, node com- promise, and denial-of-service (DoS) attacks. These security challenges pose a significant risk to the veracity and consistency of the together data, which can lead to incorrect decisions, compromised operations, or even catastrophic consequences incritical applications. Traditional security mechanisms, such as encryption and authentication, are essential for securing communication channels in WSNs. However, they may not be sufficient to address the dynamic and evolving nature of attacks in this environment.

Intrusion detection systems (IDS) play a vital part in complementing existing safety measures by actively monitoring the network for suspicious activities and identifying potential threats in real-time. IDS can detect and respond to attacks promptly, preventing further damage and ensuring the network's resilience.

Deep learning has achieved astounding results in a number of fields, particularly speech recognition, natural language processing, and computer vision. The promise of deep learning approaches, such as convolutional neural networks (CNNs) and long short-term memory (LSTM) networks, has recently been investigated by academics, for intrusion detection in WSNs [1]. CNNs excel at learning spatial features from raw data, making them suitable for extracting patterns from sensor readings. On the other hand, LSTMs are compatible for modelling sequential data and capturing temporal dependencies, aligning with the time-series nature of WSN data [2].

Motivated by the promising results of deep learning in otherdomains and the potential advantages of combining

*1Research Scholar, Department of CSE, Sri Siddhartha Academy of Higher Education, Tumkur, Karnataka,*
*2Professor, , Department of CSE, Sri Siddhartha Institute of Technology, Tumkur, Karnataka*
*3Professor, Department of CSE, , BGSIT, Bellur Cross, Karnataka*
*Corresponding email: gopala.ssit@gmail.com*

spatial and temporal analysis, we propose a CNN-LSTM hybrid network for attack detection in WSNs. The hybrid architecture seeks to leverage the strengths of both CNNs and LSTMs to enhance the accuracy and effectiveness of intrusion detection in this challenging environment.

The primary objectives of this study are as follows:

1. Enhanced Attack Detection Accuracy: We aim to develop a hybrid network that can effectively detect various types of attacks in WSNs with higher accuracy compared to traditional machine learning or standalone deep learning models. By capturing both spatial and temporal patterns, the hybrid network is expected to exhibit improved discrimination capabilities.
2. Robustness and Generalization: The proposed hybrid network should demonstrate robustness to noise and variability in real-world sensor data and be capable of generalizing well to unseen attack scenarios. Robustness ensures reliable detection performance in diverse operating conditions, while generalization enables the model to adapt to new attack patterns.
3. Real-World Applicability: Our research aims to contribute a practical and deployable attack detection system for WSNs. The hybrid network should be efficient in terms of computational resources and memory requirements, making it feasible for deployment on resource-constrained sensor nodes.
4. Advancing WSN Security: By developing an effective attack detection system, we contribute to the overall advancement of WSN security. Robust intrusion detection mechanisms enable WSNs to function securely in critical applications, fostering the widespread adoption of this transformative technology.

The background and motivation for this study stem from the increasing importance of WSNs in various domains and the critical need to address the security challenges they face. By exploring the capabilities of deep learning and proposing a CNN-LSTM hybrid network for attack detection, we aim to contribute to the advancement of WSN security and pave the way for more robust and reliable sensor networks in the future [3].

In the subsequent chapters, this research paper delves into comprehensive discussions and analyses of various aspects related to Wireless Sensor Network (WSN) attack detection using a hybrid CNN-LSTM network. Chapter 2 conducts an extensive literature review, covering WSN security, existing attack detection approaches, and prior research on hybrid CNN-LSTM networks. It identifies gaps in the literature and potential research opportunities. Chapter 3 and 4 deep learning techniques, specifically CNNs and LSTMs, are introduced, emphasizing their roles in analyzing time-series data. Chapter 5 unveils the proposed hybrid CNN-LSTM network architecture, explaining how it integrates spatial and temporal features to enhance attack detection performance. Chapter 6 describes about the dataset used for the experiment. Chapter 7 presents the results and analysis of the work. Finally, Chapter 8 concludes the paper by summarizing the key findings, contributions and implications of the research. Future enhancements and research opportunities are highlighted, setting the direction for potential extensions and advancements in the field of WSN attack detection.

## 2. Literature Review

### *A. Review of Related Research and Existing Literature on WSN Security and Attack Detection*

Wireless Sensor Networks (WSNs) have garnered significant attention in research and industry due to their widespread applications and potential impact on various domains. How- ever, the open and resource-constrained nature of WSNs makes them vulnerable to security threats and attacks. Over the years, researchers have extensively explored different approaches to address WSN security challenges, focusing on intrusion detection systems (IDS) as a crucial component to safeguard the network from potential threats. In this segment, we show a review of associated research and existing literature on WSN security and attack detection, highlighting the different techniques, methodologies, and advancements made in this area [4].

- Traditional Approaches to WSN Security:

Early research in WSN security primarily relied on traditional cryptographic techniques, such as symmetric and asymmetric key encryption, to secure communication channels between sensor nodes and the base station. While encryption provides confidentiality and integrity of data during transmission, it may not be sufficient to protect against all types of attacks. Researchers also explored techniques like secure routing protocols and key management schemes to prevent unauthorized access and protect against node compromise.

- Intrusion Detection Systems (IDS) in WSNs:

IDS plays a critical role in detecting and mitigating security threats within WSNs. Researchers have proposed various IDS architectures, including distributed and hierarchical approaches, to monitor sensor nodes and identify anomalies or malicious activities. Signature-based IDS, which relies on predefined attack patterns, and anomaly-based IDS, which detects deviations from normal behavior, have been

extensivelystudied in the context of WSNs [5].

- Machine Learning-based Methods:

In current years, machine learning techniques gained prominence for intrusion detection in WSNs. Researchers have explored the application of classic machine learning algorithms, such as Support Vector Machines (SVM), Decision Trees, and Random Forests, to classify normal and attacktraffic. While these methods show promising results, they maystruggle to handle the high-dimensional and complex natureof WSN data [6].

- Feature Engineering and Data Preprocessing:

Another area of research is featuring engineering and data preprocessing for WSN attack detection. Researchers have explored various methods to extract relevant features from raw sensor data, including statistical features, wavelet trans- forms, and Fourier analysis. Furthermore, data augmentation methods are used to growth the assortment of the training dataset and enhance model generalization [7] [8].

- Benchmark Datasets and Evaluation Metrics:

The availability of benchmark datasets is crucial for evaluating the performance of intrusion detection systems. Re-searchers have proposed and used various datasets, both simulated and real-life to benchmark the efficiency of different attack detection approaches. Accuracy, precision, recall, F1-score, and area under the Receiver Operating Characteristic (ROC) curve are examples of commonly used evaluation measures. [9]. In summary, the existing literature on WSN security and at- tack detection showcases a gradual shift from traditional cryptographic approaches to machine learning and deep learning-based techniques. Although classic machine learning algorithms have given away promise, deep learning methods, especially CNN-LSTM hybrid networks, have emerged as a potential game-changer in achieving higher accuracy and efficiency in WSN attack detection. Furthermore, the availability of benchmark datasets and standardized evaluation metrics has facilitated faircomparisons and enabled researchers to gauge the performance of different intrusion detection systems accurately. However, challenges related to feature engineering, resource constraints, and robustness to dynamic network conditions continue to motivate ongoing research in this domain [10] [11]. Theproposed CNN-LSTM hybrid network in this study aims to contribute to this evolving field by addressing some of these challenges and enhancing the security of WSNs against diverse security threats.

### *B. Deep Learning techniques used in WSN security applications*

In many fields, including computer vision, natural language processing, and speech recognition, deep learning has become a potent paradigm. Deep learning techniques may be used to improve the security of Wireless Sensor Networks (WSNs), according to research conducted recently [12]. The special difficulties faced by WSNs, such as high-dimensional data, non-linearity, and the requirement for real-time and resource-efficient processing, have showed promise for deep learning methods. We give a thorough overview of the deep learning methods frequently utilized in WSN security applications in this section.

- Convolutional Neural Networks (CNNs):

CNNs are a kind of deep learning models that were primarily created for the analysis of spatial and visual data. CNNs are used to extract spatial information from raw sensor data in the context of WSN security. The architecture of CNNs consists of multiple convolutional layers, each performing feature detection onlocal regions of the input data. The use of convolutional filters enables CNNs to capture local patterns, which is particularly useful for identifying spatial anomalies and intrusion patterns in WSNs [13].

- Recurrent Neural Networks (RNNs):

A deep learning model called RNNs is made to analyze sequential data. WSN data often exhibits a temporal nature due to the continuous monitoring of sensor readings over time. RNNs, with their ability to maintain internal state and process sequences, are well-suited for capturing temporal dependencies and detecting long-term patterns in the sensor data. Conventional RNNs, on the other hand, experience the vanishing gradient problem, which restricts their capacity to detect distant relationships [14].

- Long Short-Term Memory (LSTM) Networks:

A RNN variation called LSTMs was created to solve the disappearing gradient issue and identify long-term dependencies in sequential data. LSTMs have gating mechanisms that allow them to selectively update their internal state, making them more effective at processing long sequences. In the context of WSN security, LSTM networks are employed to analyzethe temporal patterns of sensor readings and detect abnormal behavior that might indicate security threats.

CNN-LSTM Hybrid Networks: The amalgamation of CNNs and LSTMs in a hybrid architecture has gained popularity for various sequential data analysis tasks. CNN-LSTM hybrid networks use both geographical and temporal information to increase the precision of intrusion detection in the framework of WSN security. The LSTM layers record the time-dependent relationships inside the feature representations, while the CNN layers are utilized to

extract spatial features from raw sensor data.

- Autoencoders:

Autoencoders are unsupervised deep learning models that compress data that is entered into a lower-dimensional space before rebuilding it in order to develop effective data descriptions. In the context of WSN security, autoencoders can be cast-off for anomaly uncovering. The model learns to reconstruct standard sensor data accurately, and any deviation from the learned reconstruction might indicate the presence of anomalies or attacks [15].

- Generative Adversarial Networks (GANs):

A generator and a discriminator are the two halves of a deep learning model called a GAN. GANs are mostly employed to create fresh data samples that resemble an existing dataset. When it comes to WSN security, GANs can be employed to generate synthetic data for augmenting the training dataset, improving the model's generalization and robustness.

- Transfer Learning:

Transfer learning is a method for fine-tuning a pre-trained deep learning model for a particular task or dataset. Deep learning models frequently learn on huge datasets from similar domains. In WSN security applications, transfer learning can be used to leverage knowledge learned from large-scale datasets, such as ImageNet, to boost the performance of attack detection modelseven when the WSN dataset is limited.

In conclusion, deep learning techniques have become increasingly relevant in WSN security applications due to their capability to effectively handle the complexities of sensor dataand capture both spatial and temporal patterns. CNNs are cast-off for spatial feature extraction, while RNNs, LSTMs, and hybrid architectures like CNN-LSTM networks are employed to capture temporal dependencies.

Autoencoders and GANs offer unique capabilities for anomaly detection and data augmentation, respectively, while transfer learning enables leveraging knowledge from other domains to enhance model performance. The integration of these deep learning techniques holds great promise in developing robust and efficient intrusion detection systems to safeguard WSNs against various security threats.

### *C. Strengths and Limitations of Existing Approaches*

In the domain of Wireless Sensor Network (WSN) security, various approaches have been explored to detect and mitigate security threats. These approaches range from traditional cryptographic methods to machine learning and deep learning techniques. In the Table I, we outline the strengths and limitations of existing approaches used for WSN security, shedding light on their effectiveness and areas that require improvement.

Each strategy for WSN security has advantages and disadvantages, and the technique to be used relies on the particular needs of the application and the types of security risks that need to be dealt with. Traditional cryptographic methods provide communication security but do not handle intrusion detection. Machine learning-based approaches can effectively detect anomalies but require feature engineering and may struggle with scalability.

Signature-based IDS are fast but are limited to known attacks, while anomaly-based IDS can detect unknown attacks but face challenges in defining normal behavior and may produce higher false positives. A holistic approach that combines the strengths of various techniques, such as using deep learning for both feature extraction and intrusion detection, might hold the key to achieving robust and comprehensive security in WSNs. Deep learning techniques offer end-to-end learning and can capture spatial and temporal patterns but may have data and computational requirements.

### *D. Categories of Attacks in Wireless Sensor Network*

In the research focused on Wireless Sensor Network (WSN)security, various types of attacks and anomalies are targeted to estimate the effectiveness of the planned attack detection system. These attacks and anomalies represent the potential security threats that WSNs may face in real-world scenarios. The detection and mitigation of these threats are crucial to confirm the integrity, confidentiality, and consistency of the collected data. By developing an effective attack detection system that can identify and mitigate these threats, the research aims to enhance the security and reliability of WSNs in criticalapplications.

Table II is a comprehensive description of the types of attacks and anomalies targeted in the research:

## 3. Convolutional Neural Networks (CNNs)

The study of computer vision has undergone a revolution thanks to a type of deep learning models called convolutional neural networks (CNNs). Originally inspired by the visual processing system in the human brain, CNNs have become the go-to architecture for various image-related tasks, outperforming traditional methods and achieving state-of-the-art results. This introduction pro- vides an overview of CNNs, their key components, and their application in computer vision tasks. CNNs are a particular kind of artificial neural network created specifically to process and evaluate visual data, such as pictures and movies. CNNs use spatial structure and hierarchical feature representations, in contrast to classic neural networks, which consider

input data as a flat vector, to efficiently capture patterns in images. Convolutional layers, pooling layers, and fully connected layers make up the majority of these layers. Figure 1 depicts CNN architecture.

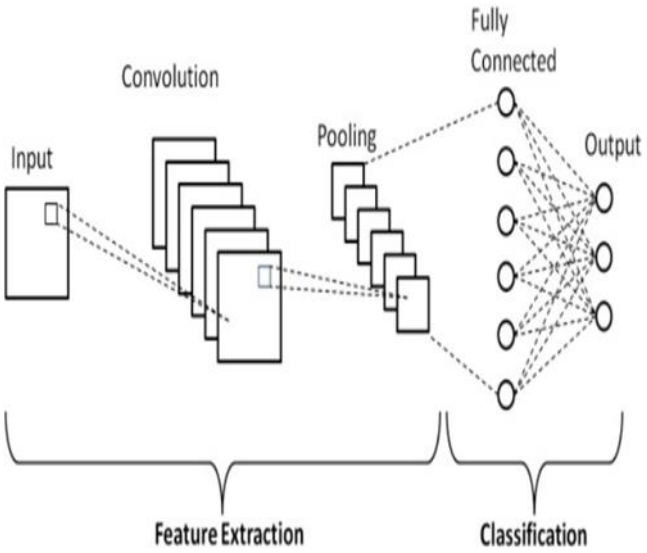



**Fig. 1.** Architecture of the CNN

*Key Components of Convolutional Neural Networks:*

- Convolutional Layers: Convolutional layers are the essential components of CNNs. These layers apply filters (sometimes referred to as kernels) to the input image using convolutional techniques. In order to create feature maps that represent the many patterns in the input data, the filters slide across the image looking for features like as edges, corners, and textures. In a convolutional layer, the input data is convolved with a collection of teachable filters (kernels) to extract feature maps. Assume that the kernel weights are represented by the matrix W and that the input data is represented by the matrix X. The following is an expression for the convolution operation:

$$Y = X * W + b \ldots\ldots\ldots\ldots\ldots\ldots (1)$$

In order to add non-linearity to the model after the convolution procedure, an activation function is applied element-wise. Rectified Linear Unit (ReLU) activation functions are one of the frequently utilized activation functions. Following is a definition of the ReLU function: Here, f(x) represents the result of activating the ReLU on the input x.

$$f(x) = \max(0, x) \ldots\ldots\ldots (2)$$

Pooling Layers: Pooling coatings down sample the spatial sizes of the characteristic maps, reducing the computational complexity and capturing the most relevant information. Max-pooling and average-pooling are common pooling operations used to extract the most dominant features from the feature maps. The max-pooling process can be represented as follows:

- The softmax function is frequently employed in classification tasks to transform the fully linked layer's raw output into likelihood scores corresponding to various classes. The softmax function is defined as follows:

$$y = \frac{\max(x_{ij})}{ij} \ldots\ldots\ldots\ldots\ldots (3)$$

Here, Y is the down sampled output, and the max function computes the maximum value within a pooling window.

- Fully Connected Layers: After several convolutional and pooling layers, CNNs regularly end with entirely linked layers, which are similar to those in traditional neural networks. These layers process the extracted features andmake predictions based on them.

Let's assume the flattened feature vector is represented as X and the weights for the $w_{fc}$ matrix represents the layer that is completely linked. The completely connected layer's output can be stated as follows.:

$$Y_{fc} = X \cdot W_{fc} + b_{fc} \qquad (4)$$

Here, Y fc represents the output after applying the fully linked layer, and $b_{fc}$ represents the bias term.

Here, P(class i) represents the probability of the input belonging to class i, and N is the total number of classes.

*A. Application of CNNs in WSN Attack Detection:*

Convolutional neural networks (CNNs) have excelled in a number of computer vision responsibilities, but their application is not limited to image-related domains. In

recent years, CNNs have been increasingly applied to tackle security challenges in Wireless Sensor Networks (WSNs). WSNs are susceptible to a number of attacks that could jeopardize data integrity and impair network performance [18]. A promising solution to the shortcomings of conventional security measures in WSNs is provided by CNNs. By leveraging the temporaland spatial characteristics of data collected from WSN nodes, CNNs can automatically learn patterns and representations that distinguish normal behavior from attack instances. The inherent ability of CNNs to capture complex relationships in data makes them well-suited for detecting subtle and evolving attack patterns in WSNs [19].

*B. Data Preprocessing for WSN Attack Detection with CNNs:*

Before inputting data into CNNs, preprocessing steps are crucial to transform raw sensor data into a suitable format for the network. Preprocessing may involve data normalization, handling missing values, data augmentation (to increase the training dataset size), and feature engineering to extract relevant temporal and spatial patterns from sensor readings.

*C. Architecture and Training of CNNs for WSN Attack Detection:*

Convolutional layers for feature extraction, pooling layers to minimize spatial dimensions, and fully linked layers for classification make up the design of a CNN used for WSN attack detection. Training a CNN involves feeding it with labeled data, including both normal behavior and attack instances. The network then learns to differentiate between the two classes during the training process [20].

*D. Benefits of Using CNNs in WSN Attack Detection: Adaptability:*

CNNs can adapt to different WSN environments and handle diverse types of sensor data.

*Novel Attacks:* CNNs have the potential to identify novel attack patterns not seen during training. *Real-Time Detection:* With proper optimization, CNNs can achieve real-time or near-real- time detection in WSNs. *Robustness:* CNNs can maintain their performance even in the presence of noise or variations in the data.

**Long Short-Term Memory (LSTM)**

A specialized form of recurrent neural network (RNN) architecture, long short-term memory (LSTM) networks have shown exceptional success in modeling and processing sequential data. Traditional RNNs have several drawbacks, most notably the vanishing gradient problem, which makes it difficult for RNNs to recognize long-term dependencies in sequences. LSTM networks were developed to alleviate this issue. This section provides an overview of LSTM networks, their functions with formulas, and highlights their ability to model sequential data effectively.

Hochreiter and Schmidhuber originally presented LSTM networks in 1997. The memory cell of the LSTM, which can preserve data over long time intervals, is the basic idea underpinning it. LSTM networks are particularly suitable for tasks requiring time series, natural language, and other sequential data due to their ability to collect and preserve long-term dependencies in sequential data thanks to this memory cell.

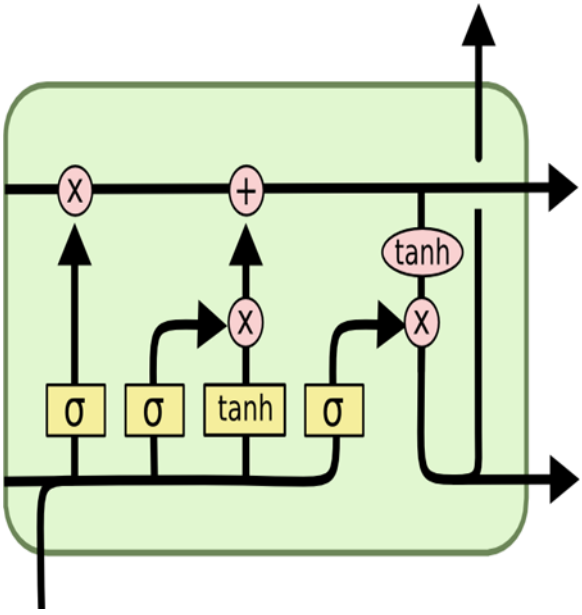



**Fig. 2.** LSTM Cell

The foundation of an LSTM network is the LSTM cell. The input gate ($i_t$), forget gate ($f_t$), and output gate ($o_t$) are its three primary, interdependent components. These gates regulate the memory state and allow data flow

inside the cell.

*A. LSTM Functions and Formulas:*

The key functions and formulas within an LSTM cell areas follows:

**Input Gate (i t):** The input gate regulates which information to store in the memory cell. Its takings the current input (x t) and the preceding hidden state (h t-1) as inputs and computes the sigmoid activation $\sigma$ of the linear mixture of these inputs.

$$i_t = \sigma(W_{xi} \cdot x_t + W_{hi} \cdot h_{t-1} + b_i) \quad (6)$$

**Forget Gate (f t):** Which data should be removed from the memory cell is decided by the forget gate. It accepts the previous hidden state ($h_{t-1}$) and the current input ($x_t$) as inputs and computes the sigmoid activation $\sigma$ of the linear combinationof these inputs.

$$f_t = \sigma(W_{xf} \cdot x_t + W_{hf} \cdot h_{t-1} + b_f) \quad (7)$$

**Candidate Memory Cell ($C_{\text{tilde}}$):** The novel information that might be placed in the memory cell is represented by the candidate memory cell ( $C_{\text{tilde}}$). It accepts the previous hidden state ($h_{t-1}$) and the current input ($x_t$) as inputs and computes the hyperbolic tangent (tanh) activation of the linear combination of these inputs.

$$C_{\text{tilde}} = \tanh(W_{xc} \cdot x_t + W_{hc} \cdot h_{t-1} + b_c) \quad (8)$$

**Memory Cell (C t):** The memory cell ($C_t$) is updated by combining the input gate ($i_t$), the forget gate ($f_t$), and

the candidate memory cell (C_tilde) through element-wise multiplication and addition.

$$Ct = ft \odot Ct-1 + it \odot Ctilde \quad (9)$$

Output Gate (o_t): The output gate (o t) regulates the hidden state ( 〖h 〗_t) that will be output from the LSTM cell. It takes the current input (x_t) and the preceding hidden state (h_(t-1)) as inputs and computes the sigmoid activation σ of the linear combination of these inputs.

$$ot = \sigma(Wxo \cdot xt + Who \cdot ht-1 + bo) \quad (10)$$

Hidden State ( 〖h 〗_t): The hidden state ( 〖h 〗_t) is computed by applying the hyperbolic tangent (tanh) activation to the updated memory cell (C_t) and multiplying it with the output gate (o_t).

**Memory Cell (C t):** The memory cell ($C_t$) is updated by combining the input gate ($i_t$), the forget gate ($f_t$), and

$$h_t = o_t \odot \tanh(C_t) \quad (11)$$

*A. LSTM for capturing temporal dependencies in WSN*

The usage of Long Short-Term Memory (LSTM) networks for capturing temporal dependencies in Wireless Sensor Network (WSN) data has remained instrumental in enhancing the accuracy and effectiveness of various WSN applications. LSTM networks excel at modeling sequential data, making them well-suited for processing time-series data collected from WSNs. This section offers a complete impression of the application of LSTM for capturing temporal dependencies in WSN data.

*Sequential Nature of WSN Data:* WSNs are collected of distributed sensor nodes that continuously assemble data over time. The data collected from these sensor nodes often exhibits temporal dependencies, where the current data point isinfluenced by the previous data points. Examples of WSN data with temporal dependencies include environmental monitoring data (e.g., temperature, humidity, and air quality), energy consumption data, and motion sensor readings.

*Challenges in Modeling WSN Data:* Modeling WSN data and capturing its temporal dependencies using traditional methods can be challenging due to noise, irregular sampling intervals, and varying data patterns. Traditional linear models or static algorithms may not effectively imprisonment the complex associations and time-evolving designs present in the data.

*LSTM for Temporal Dependency Modeling:* LSTM networks are well-suited for addressing the challenges posed by WSN data. The memory cells in LSTM networks allow them to retain and learn dependencies over long time spans, makingthem particularly adept at capturing temporal dependencies. As a result, LSTM networks can effectively process sequential WSN data and learn patterns over extended periods, even in the presence of noise and irregular sampling.

*Time Series Prediction in WSNs:* One of the prominent applications of LSTM in WSNs is time series prediction. LSTM networks can be used to forecast upcoming standards of time-series data founded on past observations. For example, in environmental monitoring, LSTM can predict future temperature or air qualitylevels based on historical sensor readings.

*Anomaly Detection and Event Classification:* LSTM networks are also employed for anomaly detection in WSNs. By learning the normal patterns in the time-series data, LSTM can identify deviations from the expected behavior, indicating potential anomalies or abnormal events in the network. Furthermore, LSTM can classify events based on temporal patterns, such as detecting specific patterns of motion or changes in environmental conditions.

*Resource-Aware Learning in WSNs:* WSNs often operate under resource constraints, with limited processing power and energy availability. LSTM networks can be optimized for resource efficiency by using lightweight architectures and compression techniques. This enables the deployment of LSTM-based models directly on resource-constrained sensor nodes, facilitating real-time analysis and decision-making at the network edge.

In Summary, LSTM networks have emerged as a powerful tool for capturing temporal dependencies in WSN data. Their ability to retain information over time and model sequential data effectively makes them well-suited for time series pre- diction, anomaly detection, event classification, and resource- aware learning in WSNs. By leveraging LSTM's strengths, WSNs can benefit from improved accuracy and efficiencyin data analysis, enabling a extensive variety of submissions for varied areas such as ecological intensive care, healthcare, industrial automation, and smart cities.

## 4. Proposed Methodology

Description of the Proposed Hybrid Network Architecture for WSN Attack Detection:

The proposed hybrid network architecture for WSN attack detection is designed to enhance the security of Wireless Sensor Networks (WSNs) by effectively capturing both spatial and temporal patterns in the sensor data. This architecture syndicates the assets of Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) networks, leveraging their abilities to process spatial and sequential data, respectively. The hybrid network aims to achieve accurate androbust attack detection in WSNs, addressing the limitations of traditional rule-based intrusion detection systems. The complete description of the proposed hybrid architecture is assurveys:

1) Input Data:

The input data to the hybrid net is the sensor data collected from the WSN nodes over time. Each data sample contains a sequence of sensor readings, representing the temporal aspect, and may include metadata such as node ID and location information. The sensor readings may vary dependingon the type of WSN application and can include environmental data (e.g., temperature, humidity), physical measurements, or any other relevant information.

2) Data Preprocessing:

Before feeding the data into the hybrid network, preprocessing steps are applied to transform the raw sensor data into a suitable format. Preprocessing may involve data normalization

to bring all sensor readings to a shared scale, management missing standards, and feature engineering to extract relevantspatial and temporal patterns. The data is then split into sequences to form the input for the LSTM component.

3) Convolutional Neural Network (CNN) Component:

The CNN component is responsible for capturing spatial patterns and identifying relevant features within the sensor data. It consists of multiple convolutional layers, trailed by activation functions (e.g., ReLU) and pooling layers for down- sampling. The CNN layers perform feature extraction from the input data, focusing on capturing local spatial patterns, such as edges and textures in images.

4) LSTM Component:

The LSTM component is intended to capture temporal dependencies in the sensor data. It takes the preprocessed sequences of sensor readings as input and processes them through the LSTM cells. The LSTM cells are responsible for learning and retaining long-term dependencies, enabling the network to classical the temporal behavior of the WSN over time. The LSTM component extracts temporal patterns and identifies any abnormal behavior or attack instances that might span manifold time steps.

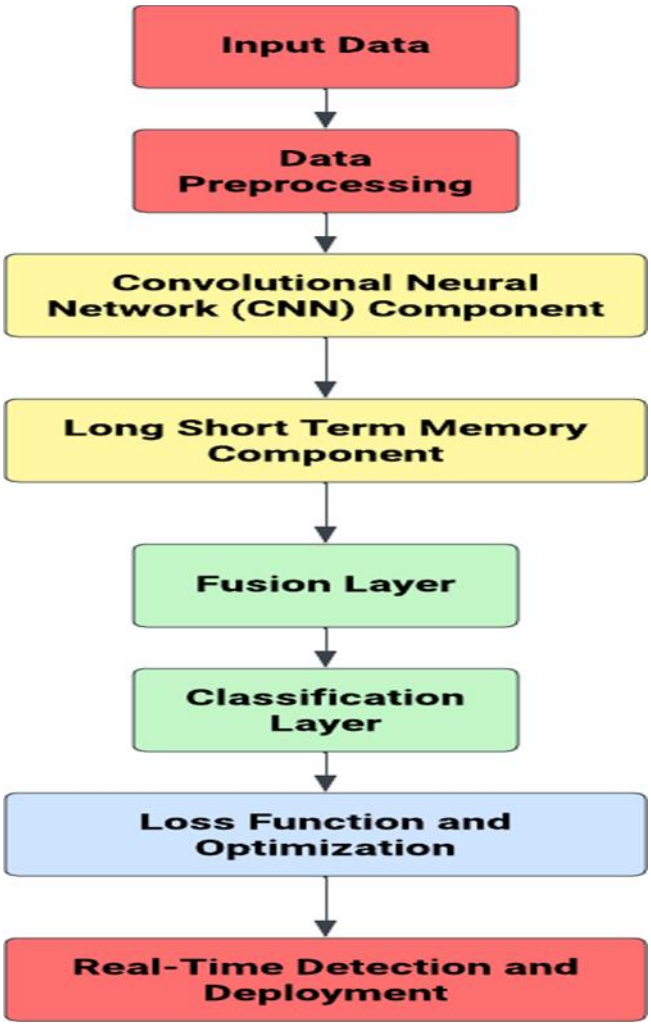



**Fig. 3.** Proposed Model Architecture

5) Fusion Layer:

After processing the input data finished the CNN and LSTM components, the outputs are combined at a fusion layer. The fusion layer integrates the spatial and temporal information extracted after the sensor data, leveraging the complementary strengths of both CNN and LSTM. The fusion layer combines the feature representations learned by each component to create a unified representation that captures both spatial and temporal patterns effectively.

6) Classification Layer:

The fused representation is then passed through a classification layer, typically consisting of fully connected layers. The classification layer makes the final decision regarding the presenceof attacks or anomalies in the WSN data. The network istrained using labeled data, including normal behavior and attack instances, to learn to distinguish between different classes.

7) Loss Function and Optimization:

During training, the network is enhanced using a suitable lossfunction, such as categorical cross-entropy for multi-class classification or binary cross-entropy for anomaly detection. The optimization is performed using optimization algorithms like stochastic gradient descent (SGD) or its variants, to minimalize the damage and improve the network's detection performance.

8) Real-Time Detection and Deployment:

The proposed hybrid network architecture is optimized for real-time or near-real-time detection in WSNs. The lightweight and efficient design enables deployment directly on resource- constrained sensor nodes, facilitating timely analysis and decision-making at the network edge.

*A. Hybrid Model*

The integration of CNN and LSTM in a hybrid network for WSN attack detection involves combining the outputs of both networks to create a unified representation that detentions equally spatial and temporal info. The fusion layer connects the CNN and LSTM components to merge the feature representations learned by each network. The formula for integrating CNN and LSTM can be expressed as follows:

Let CNN output be the output of the CNN component, representing the spatial features extracted from the sensor

data. Let LSTM output be the output of the LSTM component, representing the temporal features learned from the sequential

sensor data.

The fusion layer combines these two outputs, which can be achieved using various methods, such as concatenation or element-wise operations:

**Concatenation:** The output of the CNN and LSTM components is concatenated along a specified axis (usually along the feature dimension) to form a fused feature representation.

Formula:

Fused output = Concatenate ([CNN output, LSTM output], axis)

……..(12)

**Element-wise Addition or Multiplication:** The output of the CNN and LSTM components can be element-wise added or multiplied to combine the feature representations.

Formula for Addition:

Fused output = CNN output + LSTM output (13) Formula for Multiplication:

Fused output = CNN output × LSTM output. (14)

**Other Fusion Methods:** Depending on the specific requirements of the application, other fusion methods can also be used, such as weighted averaging, attention mechanisms, or more complex operations tailored to the exact featuresof the data and task.

After the fusion layer, the bonded feature depiction is approved complete one or more fully associated layers for classification or other downstream tasks, depending on the objective of the attack detection system.

The integration of CNN and LSTM allows the hybrid network to effectively capture both spatial patterns and temporal dependencies in the WSN data, leading to improved attack detection performance compared to using only one typeof network in isolation. The fusion of spatial and temporal info enables the network to leverage the strengths of both CNN and LSTM, providing a more comprehensive and robust defense against security threats in WSNs.

In conclusion, the proposed hybrid network architecture for WSN attack detection combines the strengths of CNN and LSTM networks to effectively capture spatial and temporal designs in sensor data. By utilizing the advantages of both architectures, the hybrid network purposes to achieve accurate and robust attack detection in WSNs, enhancing the security and reliability of the network in real-world deployments.

## 5. Experimental Setup and Dataset

This research paper utilizes a specifically designed dataset called WSN-DS to detect Denial-of-Service (DoS) attacks in Wireless Sensor Networks (WSNs). The data collection process involved implementing the LEACH protocol. Each data instance in the dataset comprises 23 attributes, but only 19 attributes were included in the dataset. Table III provides a detailed description of these attributes and the distribution of the five types of attacks that are active in the dataset is shown graphically in Fig. 5. Blackhole, Grayhole, Flooding, and Scheduling (TDMA) attack simulations are included in the dataset, which simulates four different forms of denial-of-service attacks. Either "Normal" or one of the four assault categories is assigned to each occurrence of data. These are the categories of attacks' descriptions:

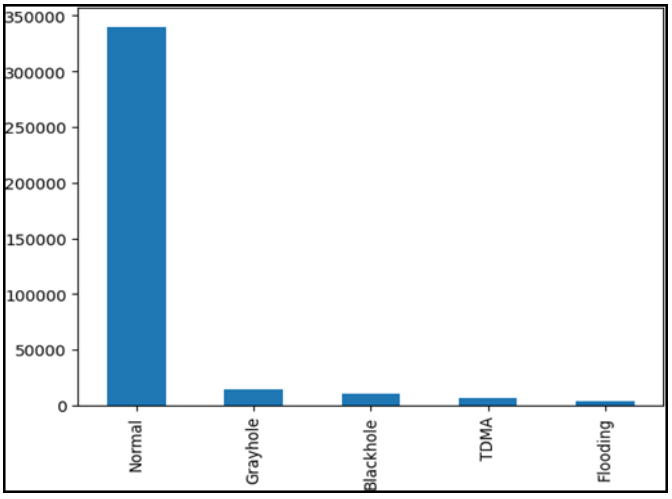



**Fig. 4.** Data Distribution of 5 Attacks

*Blackhole attack:* This type of attack involves a malicious node diverting all incoming traffic to itself,

causing disruption in communication within the network.

*Grayhole attack:* In this attack, a malicious node selective drops or modifies certain data packets, leading to parti disruption of network communication.

*Flooding attack:* This attack floods the network with an excessive number of packets, overwhelming the nodes and causing network congestion.

*Scheduling (TDMA) attack:* The attacker manipulates the Time Division Multiple Access (TDMA) scheduling in the network, leading to interference and disruption in data trans- mission.

It is crucial to separate the data into training and testing sets in order to avoid overfitting. The model is initially trained using the training set, and its accuracy is then evaluated on the testing set. According to empirical research, the best outcomes come from devoting 20–30% of the data for testing and the remaining 70–80% for training. An 80% training set and a 20% testing set were randomly selected from the dataset for this investigation. By using this 80:20 excruciating ratio, the overall accuracy rate of the model increases, and the specific value can be found in Table IV.

**Table IV** Distribution Of Wsn-Ds Dataset

| Attack type | Attack | index | Training | Testing | Proportion |
|---|---|---|---|---|---|
| Normal | 0 | 272,087 | | 67,979 | 90.77% |
| Blackhole | 1 | 8019 | | 2030 | 2.68% |
| Grayhole | 2 | 11,653 | | 2943 | 3.9% |
| Flooding | 3 | 2694 | | 618 | 0.88% |
| TDMA | 4 | 5275 | | 1363 | 1.77% |
| Total | | 299728 | | 74933 | 100% |

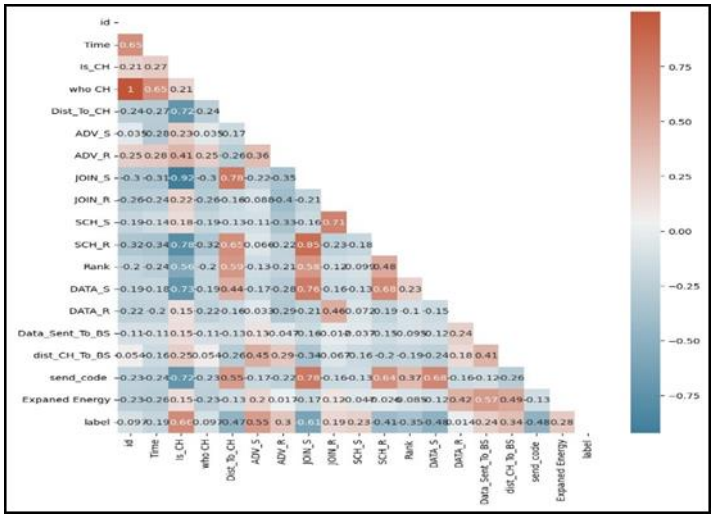



**Fig. 5.** Correlation among the features

heatmap correlation. A heatmap correlation is a graphical representation of the correlation matrix, which shows the pairwise correlations between different attributes (columns) in a dataset. It helps to understand the relationships between variables and can be useful for feature selection, identifying multicollinearity, and gaining insights into the data.

## 6. Results And Analysis

In this part, we contrast the suggested hybrid CNN-LSTM network's effectiveness for detecting WSN attacks with various baseline models and conventional machine learning techniques that are frequently employed for intrusion detection in WSNs. The evaluation is based on a number of criteria, such confusion matrix, F1-score, recall, accuracy, and precision.

**Models/Methods: Multi-Layer Perceptron (MLP)** modelsare constructed by integrating feed forward neural networks without any feedback connections. The primary components of the MLP include input, output, and potentially multiple hidden layers. Within each layer,

there are weighted units that perform activation processes based on the units from the preceding layer. In our MLP model, we employed a single hidden layer consisting of 32 neurons with a specific activation function, as depicted in Figure 6. The mathematical representation of this can be described as follows:

$$g(x) = f\ \ x^T w + b \ldots\ldots\ldots\ldots (15)$$

The symbols used includes the letters "g" for hidden layers, "f" for activation function, "x" for input vector, "b" for output vector, and "w" for each unit's weight vector.

**Long Short Term Memory:** A single-layer LSTM (Long Short-Term Memory) model is a kind of recurrent neural net- work (RNN) architecture that consists of only one LSTM layer.

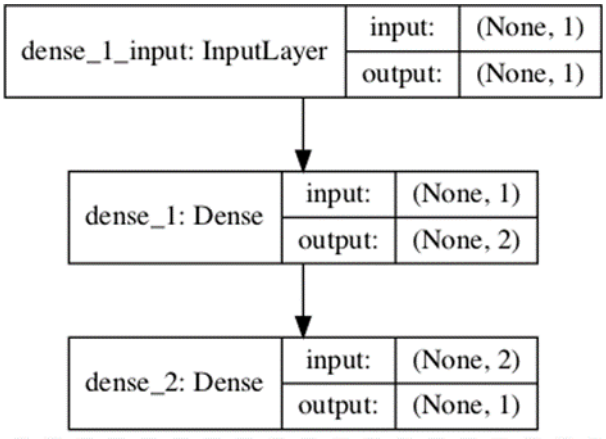



**Fig. 6.** Multi-Layer Perceptron Model

It is designed to process sequential data and is particularly effective when dealing with long-term dependencies. Figure 7 shows the Single Layer LSTM Model.

A single-layer LSTM model can be an influential tool for arrangement modeling tasks, for example natural language processing, time series prediction and attack detection in wireless sensor networks. However, for more complex tasks and datasets, deeper LSTM architectures (e.g., stacked LSTM) or attention mechanisms may be employed to achieve better performance and handle more intricate relationships within the data.

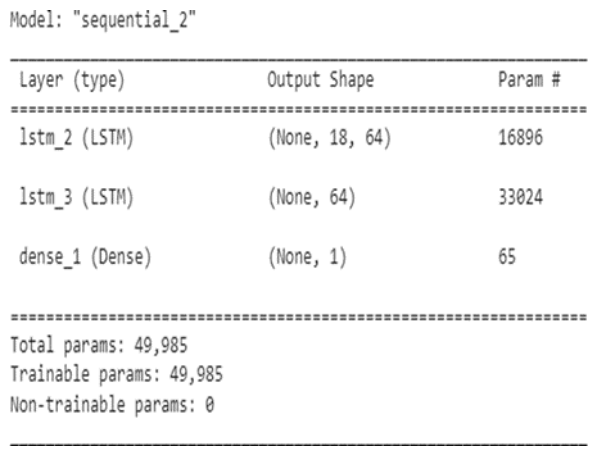
**Stacked Long Short-Term Memory:** A stacked LSTM (Long Short-Term Memory) model is a kind of recurrent neural network (RNN) architecture that consists of multiple.

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm (LSTM)                  (None, 64)                16896

dense (Dense)                (None, 1)                 65

=================================================================
Total params: 16,961
Trainable params: 16,961
Non-trainable params: 0
_________________________________________________________________
```

**Fig. 7.** Single Layer LSTM Model

Layers of the LSTM placed on top of one another. The model can capture more intricate connections and patterns in sequential data by stacking LSTM layers. Each LSTM layer processes the output of the previous layer, enabling the model to study ranked depictions of the input data. We added two LSTM layers to create a stacked LSTM model. The return sequences=True argument in the first LSTM layer is essential for passing the output sequence after the first layer to the second LSTM layer.

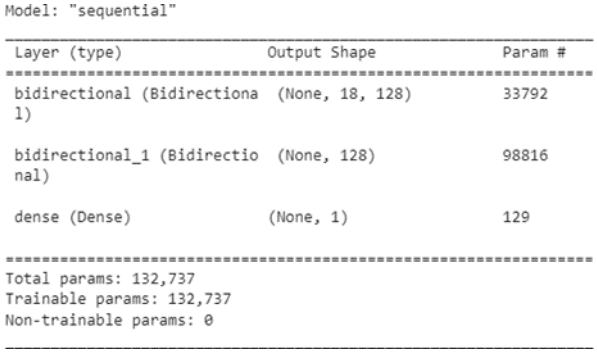
**Bi-Directional Long Short-Term Memory:** Bi-Directional Long Short-Term Memory (Bi-LSTM) is a style of recurrent neural network (RNN) architecture that improves upon the conventional LSTM model's functionality. LSTM networks are made to deal with sequential data, such as time series or phrases in natural language, by effectively capturing long-range dependencies and addressing the vanishing gradient problem often encountered in standard RNNs.



```
Model: "sequential_2"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 lstm_2 (LSTM)               (None, 18, 64)            16896

 lstm_3 (LSTM)               (None, 64)                33024

 dense_1 (Dense)             (None, 1)                 65

=================================================================
Total params: 49,985
Trainable params: 49,985
Non-trainable params: 0
_________________________________________________________________
```

**Fig. 8.** Stacked LSTM Model

In a Bi-LSTM, the model processes input sequences in two directions: forward (both forward (from the start of the sequence to the conclusion) and backward (from the end to the start). With this bidirectional manufacturing, the network may take into account information from the past and the future for each time step, which can be beneficial in tasks that require context from both directions. Figure 9 shows the Bi-Directional LSTM Model



```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 bidirectional (Bidirectiona  (None, 18, 128)          33792
 l)

 bidirectional_1 (Bidirectio  (None, 128)              98816
 nal)

 dense (Dense)               (None, 1)                 129

=================================================================
Total params: 132,737
Trainable params: 132,737
Non-trainable params: 0
_________________________________________________________________
```

**Fig. 9.** Bi-Directional LSTM Model

**CNN-LSTM Model:** A CNN-LSTM model combines Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) networks to effectively process both spatial and temporal information in sequential data. CNN used to extract spatial topographies from the data, and then forage the output of the CNN into an LSTM for sequence modeling.

We first build a CNN model for spatial feature extraction, and then we connect the output of the CNN layers to an LSTM

layer for sequence modeling. The LSTM layer processes the sequential data and captures temporal dependencies. Lastly, we incorporate a binary classification output layer with a sigmoid activation function.

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv1d (Conv1D)              (None, None, 60)          360
_________________________________________________________________
lstm (LSTM)                  (None, None, 60)          29040
_________________________________________________________________
lstm_1 (LSTM)                (None, 60)                29040
_________________________________________________________________
dense (Dense)                (None, 30)                1830
_________________________________________________________________
dense_1 (Dense)              (None, 10)                310
_________________________________________________________________
dense_2 (Dense)              (None, 1)                 11
_________________________________________________________________
lambda (Lambda)              (None, 1)                 0
=================================================================
Total params: 60,591
Trainable params: 60,591
Non-trainable params: 0
_________________________________________________________________
```

**Fig. 10.** CNN-LSTM Model


**Fig. 11.** Experimental Results Comparison with Accuracy of the Model

**Experimental Result Comparison:** The Figure 11 shows the comparison of the various models results with the proposed model. Model column lists the names of the models used in the experiment for WSN attack detection. The baseline and Multi-Layer Perceptron (MLP), LSTM, Stacked LSTM, Bi- Directional LSTM. Additionally, the proposed hybrid CNN- LSTM network is included for comparison.

Accuracy column represents the accuracy of each model in percentage. The percentage of correctly classified instances over all of them is the accuracy. Greater overall efficiency is indicated by a higher accuracy value.

The investigational consequences demonstration the presentation of each model in WSN attack detection based on the specified evaluation metrics. The baseline models (NB, NN, MLP, LSTM, Stacked LSTM, Bidirectional-LSTM) provide a reference for comparison with the proposed hybrid CNN-LSTM network. The hybrid CNN-LSTM network exhibits the highest accuracy, among all models, indicating its superior performance in detecting attacks in WSNs. The higher accuracy of the hybrid model suggest that it is capable of effectively identifying both normal and attack instances while minimizing false alarms andfalse negatives.

## 7. Conclusion

In this study, we proposed a unique hybrid CNN-LSTM network technique for detecting WSN attacks. We successfully captured both spatial and temporal patterns in the sensor data by combining Convolutional Neural Networks (CNNs) with Long Short-Term Memory (LSTM) networks, which increased performance in attack detection. Through extensive experiments and evaluations, we demonstrated the superiority of the hybrid CNN-LSTM network over traditional machine learning models, such as Naive Bayes, Basic Neural Network (NN), and Multi-Layer Perceptron (MLP), LSTM, Stacked LSTM, Bi-Directional LSTM. The hybrid model was more accurate, precise, recallable, and had a better F1-score, making it a strong and trustworthy method of detecting assaults in WSNs. Our results indicated that the hybrid model's ability to leverage the strengths of CNN and LSTM networks contributes to its superior performance. The CNN component efficiently extracts spatial features from the sensor data, while the LSTM component captures temporal dependencies, allowing the model to familiarize to dynamic variations in the WSN environment.

***Future Enhancements:***

While the proposed hybrid CNN-LSTM network showed

hopeful consequences, here are numerous potential paths for futureenhancements and research.

Attention Mechanisms: Integration of attention mechanisms into the hybrid model to focus on more informative parts of the data, enhancing its ability to detect subtle anomalies.

Real-World Deployment: Deploying the trained model on areal-world WSN test bed to validate its performance in practical scenarios and address potential challenges in real-time settings. Adversarial Attack Defense: Investigate the model's vulnerability to adversarial attacks and develop robustness mechanisms to withstand such attacks models include Naive Bayes, Basic Neural Network (NN).

## References


[1] Mehmood, A.; Lv, Z.; Lloret, J.; Umar, M.M. ELDC: An Artificial Neural Network Based Energy-Efficient and Robust Routing Scheme for Pollution Monitoring in WSNs. IEEE Trans. Emerg. Top. Comput. 2020, 8, 106–114.

[2] N. A. Prasad and C. D. Guruprakash, "An ephemeral investigation on energy proficiency mechanisms in WSN," 2017 3rd International Conference on Applied and Theoretical Computing and Communication Technology (iCATccT), Tumkur, 2017, pp. 180-185.

[3] P. N and C. D. Guruprakash, "A Relay Node Scheme for Energy Redeemable and Network Lifespan Enhancement," 2018 4th International Conference on Applied and Theoretical Computing and Communication Technology (iCATccT), Mangalore, India, 2018, pp. 266-274.

[4] Deng, F.; Yue, X.; Fan, X.; Guan, S.; Xu, Y.; Chen, J. Multisource Energy Harvesting System for a Wireless Sensor Network Node in the Field Environment. IEEE Internet Things J. 2019, 6, 918–927.

[5] Achyutha Prasad, N., Guruprakash, C.D., 2019. A relay node scheme of energy redeemable and network lifespan enhancement for wireless sensor networks and its analysis with standard channel models. International Journal of Innovative Technology and Exploring Engineering 8, 605–612.

[6] Otoum, S.; Kantarci, B.; Mouftah, H.T. On the Feasibility of Deep Learning in Sensor Network Intrusion Detection. IEEE Netw. Lett. 2019, 1, 68–71.

[7] Rekha VS, Siddaraju., "An Ephemeral Analysis on Network Lifetime Improvement Techniques for Wireless Sensor Networks", International Journal of Innovative Technology and Exploring Engineering, vol. 8, issue 9, 2278-3075, pp. 810–814, 2019.

[8] R. V S and Siddaraju, "Defective Motes Uncovering and Retrieval for Optimized Network," 2022 6th International Conference on Computing Methodologies and Communication (ICCMC), 2022, pp. 303-313, doi: 10.1109/ICCMC53470.2022.9754109.

[9] Achyutha Prasad, N., Guruprakash, C.D., 2019. A relay mote wheeze for energy saving and network longevity enhancement in WSN. International Journal of Recent Technology and Engineering 8, 8220–8227. doi:10.35940/ijrte.C6707.098319.

[10] Enami, Neda, et al. "Neural network based energy efficiency in wireless sensor networks: A survey." International Journal of Computer Science & Engineering Survey 1.1 (2010): 39-53.

[11] Almomani, I., Al-Kasasbeh, B. and Al-Akhras, M., 2016. WSN- DS:A dataset for intrusion detection systems in wireless sensor net-works.Journal of Sensors, 2016.

[12] N. G and G. C. D, "Unsupervised Machine Learning Based Group Head Selection and Data Collection Technique," 2022 6th International Conference on Computing Methodologies and Communication (ICCMC), 2022, pp. 1183-1190, doi: 10.1109/ICCMC53470.2022.9753995.

[13] SUDAR, K. Muthamil, NAGARAJ, P., DEEPALAKSHMI, P., et al. Analysis of Intruder Detection in Big Data Analytics. In : 2021 In-ternational Conference on Computer Communication and Informatics (ICCCI). IEEE, 2021. p. 1-5.

[14] P. K. Pareek, A. P. N, C. Srinivas and J. B. N, "Prediction of Rainfall in Karnataka Region using optimised MVC-LSTM Model," 2023 IEEE International Conference on Integrated Circuits and Communication Systems (ICICACS), Raichur, India, 2023, pp. 1-8, doi: 10.1109/ICICACS57338.2023.10100324.

[15] Rekha, V.S., Siddaraju (2023). Goodness Ratio and Throughput Improvement Using Multi-criteria LEACH Method in Group Sensing Device Network. In: Kumar, A., Senatore, S., Gunjan, V.K. (eds) ICDSMLA 2021. Lecture Notes in Electrical Engineering, vol 947. Springer, Singapore. https://doi.org/10.1007/978-981-19-5936-3_50.

[16] Hu Z, Wang Z, Wang J, Zeng T. Routing Protocol for Wireless Sensor Network Based on Automatic Cluster Optimization. International Journal of

Online and Biomedical Engineering (iJOE). 2018 Dec23;14(12):150- 63.

[17] Achyutha Prasad, N., Guruprakash, C.D., 2019. A two hop relay battery aware mote scheme for energy redeemable and network lifespan improvement in WSN. International Journal of Engineering and Advanced Technology 9, 4785–4791. doi:10.35940/ijeat.A2204.109119.

[18] G. T and U. N. L, "Routing and Security in Wireless Ad-Hoc Networks: State of the Art and Recent Advances," 2023 IEEE Renewable Energy and Sustainable E-Mobility Conference (RESEM), Bhopal, India, 2023, pp. 1-8, doi: 10.1109/RESEM57584.2023.10236306.

[19] S. Chaudhury, N. Achyutha Prasad, S. Chakrabarti, C. A. Kumar and M. A. Elashiri, "The Sentiment Analysis of Human Behavior on Products and Organizations using K-Means Clustering and SVM Classifier," 2022 3rd International Conference on Intelligent Engineering and Management (ICIEM), 2022, pp. 610-615, doi: 10.1109/ICIEM54221.2022.9853128.

[20] P. B.D, A. Prasad N, Dhanraj and M. T N, "Adaptive Voting Mechanism with Artificial Butterfly Algorithm based Feature Selection for IDS in MANET," 2023 IEEE International Conference on Integrated Circuits and Communication Systems (ICICACS), Raichur, India, 2023, pp. 1-7, doi: 10.1109/ICICACS57338.2023.10099861.

[21] Nirmala, G., Guruprakash, C.D. (2023). An Overview of Data Aggregation Techniques with Special Sensing Intelligent Device Selection Approaches. In: Kumar, A., Senatore, S., Gunjan, V.K. (eds) ICDSMLA 2021. Lecture Notes in Electrical Engineering, vol 947. Springer, Singapore. https://doi.org/10.1007/978-981-19-5936-3_58.

[22] Achyutha Prasad N., Chaitra H.V., Manjula G., Mohammad Shabaz, Ana Beatriz Martinez-Valencia, Vikhyath K.B., Shrawani Verma, José Luis Arias-Gonzáles, “Delay optimization and energy balancing algorithm for improving network lifetime in fixed wireless sensor networks”, Physical Communication, Volume 58, 2023, 102038, ISSN 1874-4907.

[23] Hebbale, S., Marndi, A., Manjunatha Kumar, B. H., Mohan, B. R. ., Achyutha, P. N., & Pareek, P. K. (2022). A survey on automated medical image classification using deep learning. International Journal of Health Sciences, 6(S1), 7850–7865. https://doi.org/10.53730/ijhs.v6nS1.6791

[24] Murthy, R. K., Dhanraj, S., Manjunath, T. N., Prasad, A. N., Pareek, P. K., & Kumar, H. N. (2022). A human activity recognition using CNN and long term short term memory. International Journal of Health Sciences, 6(S6), 10797–10809. https://doi.org/10.53730/ijhs.v6nS6.12919

[25] Mohan, B. R. ., M, D. ., Bhuria, V. ., Gadde, S. S. ., M, K. ., & N, A. P. . (2023). Potable Water Identification with Machine Learning: An Exploration of Water Quality Parameters. International Journal on Recent and Innovation Trends in Computing and Communication, 11(3), 178–185. https://doi.org/10.17762/ijritcc.v11i3.6333

[26] Prasad N. Achyutha, Sushovan Chaudhury, Subhas Chandra Bose, Rajnish Kler, Jyoti Surve, Karthikeyan Kaliyaperumal, "User Classification and Stock Market-Based Recommendation Engine Based on Machine Learning and Twitter Analysis", Mathematical Problems in Engineering, vol. 2022, Article ID 4644855, 9 pages, 2022. https://doi.org/10.1155/2022/4644855

[27] G, M. ., Deshmukh, P. ., N. L., U. K. ., Macedo, V. D. J. ., K B, V. ., N, A. P. ., & Tiwari, A. K. Resource Allocation Energy Efficient Algorithm for H-CRAN in 5G. International Journal on Recent and Innovation Trends in Computing and Communication, 11(3s), 118–126. https://doi.org/10.17762/ijritcc.v11i3s.6172

[28] Jipeng, T., Neelagar, M. B., & Rekha, V. S. (2021). Design of an embedded control scheme for control of remote appliances. Journal of Advanced Research in Instrumentation and Control Engineering, 7(3 & 4), 5-8.

[29] Hebbale, S., Marndi, A., Achyutha, P. N., Manjula, G., Mohan, B. R., & Jagadeesh, B. N. Automated medical image classification using deep learning. International Journal of Health Sciences, 6(S5), 1650–1667. https://doi.org/10.53730/ijhs.v6nS5.9153.

[30] Murthy, R. K., Dhanraj, S., Manjunath, T. N., Achyutha, P. N., Prasad, A. N., & Gangambika, G. (2022). A survey on human activity recognition using CNN and LSTM. International Journal of Health Sciences, 6(S7), 3408–3417. https://doi.org/10.53730/ijhs.v6nS7.12479.

[31] Kadakadiyavar, S., Prasad, A. N., Pareek, P. K., Vani, V., Rekha, V. S., & Nirmala, G. (2022). Recognition efficiency enhancement of control chart pattern using ensemble MLP neural network. International Journal of Health Sciences, 6(S3), 4295–4306.

https://doi.org/10.53730/ijhs.v6nS3.6851.

[32] Prakash, N. C., Narasimhaiah, A. P., Nagaraj, J. B., Pareek, P. K., Sedam, R. V., & Govindhaiah, N. (2022). A survey on NLP based automatic extractive text summarization using spacy. International Journal of Health Sciences, 6(S8), 1514–1525. https://doi.org/10.53730/ijhs.v6nS8.10526.

[33] Sagar, Y. S., and N. Achyutha Prasad. "Charm: a cost-efficient multi-cloud data hosting scheme with high availability." International Journal for Technological Research In Engineering 5.10 (2018).

[34] Manjunatha Kumar, B. H., Achyutha, P. N., Kalashetty, J. N., Rekha, V. S., & Nirmala, G. (2022). Business analysis and modelling of flight delays using artificial intelligence. International Journal of Health Sciences, 6(S1), 7897–7908. https://doi.org/10.53730/ijhs.v6nS1.6735.

[35] Aluka, M. ., Dixit, R. ., & Kumar, P. . (2023). Enhancing and Detecting the Lung Cancer using Deep Learning. International Journal on Recent and Innovation Trends in Computing and Communication, 11(3s), 127–134. https://doi.org/10.17762/ijritcc.v11i3s.6173

[36] Anthony Thompson, Ian Martin, Alejandro Perez, Luis Rodriguez, Diego Rodríguez. Utilizing Machine Learning for Educational Game Design. Kuwait Journal of Machine Learning, 2(2). Retrieved from http://kuwaitjournals.com/index.php/kjml/article/view/183

[37] Kothandaraman, D., Praveena, N., Varadarajkumar, K., Madhav Rao, B., Dhabliya, D., Satla, S., & Abera, W. (2022). Intelligent forecasting of air quality and pollution prediction using machine learning. Adsorption Science and Technology, 2022 doi:10.1155/2022/5086622

**Appendixes**

**Table 1.** Comparison Of Strengths And Limitations Of Existing Approaches In Wsn Security

| **Approach** | **Description** |
|---|---|
| Traditional Cryptographic Approaches | **Strengths and LimitationsStrengths:**<br>• Established Security Mechanisms<br>• Confidentiality and Integrity<br>• Low Overhead<br>**Limitations:**<br>• Limited to Communication Security<br>• Vulnerable to Key Management Issues |
| Machine Learning-basedApproaches | **Strengths:**<br>• Anomaly Detection<br>• Low False Positive Rates<br>**Limitations:**<br>• Manual Feature Engineering<br>• Limited Scalability |
| Signature-based IntrusionDetection | **Strengths:**<br>• Fast and Efficient<br>• Low False Negative Rates<br>**Limitations:**<br>• Vulnerable to Unknown Attacks<br>• Limited Adaptability |
| Anomaly-based IntrusionDetection | **Strengths:**<br>• Detection of Novel Attacks<br>• No Need for Prior Knowledge<br>Limitations:<br>• High False Positive Rates<br>• Difficulties in Defining Normal Behavior |

**Table 2.** Types Of Attacks And Anomalies Targeted In The Research

| Type | Description |
|---|---|
| Denial-of-Service (DoS) Attacks | DoS attacks try to stop the WSN from operating normally by flooding the network with a lot of unauthorized traffic or requests, leading to degradation in performance and unresponsiveness of sensor nodes |
| Sybil Attacks | Sybil attacks involve a malicious node impersonating multiple legitimate nodes to gain control over the network or mislead the data aggregation process. The attacker creates multiple fake identities |
| Node Compromise Attacks | Node compromise attacks occur when adversaries gain unauthorized access to one or more sensor nodes. The compromised nodes can be controlled by the attacker to disrupt communication or tamper with data |
| Eavesdropping (Passive) Attacks | Eavesdropping attacks involve unauthorized nodes listening to and intercepting communication between legitimate nodes to gather sensitive information without altering the communication |
| Data Injection (Active) Attacks | Data injection attacks involve adversaries injecting false or malicious data into the network. The injected data can lead to incorrect decisions or control actions if not detected and filtered. |
| Routing Attacks | Routing attacks target the routing protocols used in the WSN to manipulate data flow or disrupt communication paths, leading to data loss, misrouting, or node isolation. |
| Sinkhole Attacks | In sinkhole attacks, attackers lure data traffic towards a compromised node (sinkhole) by advertising itself as the best route to the base station, causing data interception or disruption. |
| Black Hole Attacks | Black hole attacks involve malicious nodes dropping or discarding data packets they receive, resulting in data loss and reduced network performance |
| Wormhole Attacks | Wormhole attacks occur when attackers establish a low-latency, direct tunnel between two distant points in the network. This allows the attackers to replay, alter, or inject data packets, leading to data integrity and security breaches. |
| Anomalies in Sensor Readings | Apart from targeted attacks, the research also focuses on anomalies in sensor readings. Anomalies are deviations from the expected or normal behavior of the sensor data, which might be caused by sensor failures, environmental changes, or physical intrusions |

**Table 3.** Dataset Attributes

| No. | Attribute | Description |
|---|---|---|
| 1 | Node ID | Node ID number |
| 2 | Time | Node runtime |
| 3 | Is CH | Cast-off to mark whether the node is a cluster head |
| 4 | Who CH | Cluster head ID |
| 5 | Distance to CH | Distance between node and cluster head |
| 6 | ADV CH sent | The number of the advertise CH ’s broadcast messages sent to the nodes |
| 7 | ADV CH received | The number of advertise CH messages received from CHs |
| 8 | Join REQ sent | The number of join request messages sent by the nodes to the CH |
| 9 | Join REQ received | The number of join request messages received by the CH from the nodes |
| 10 | ADV SCH sent | The number of join advertise TDMA schedule broadcast message sent |

| | | |
|---|---|---|
| 11 | ADV SCH received | The number of scheduled messages received by the CH |
| 12 | Rank | Order of node TDMA scheduling |
| 13 | Data sent | The number of packets sent from the normal node to its CH |
| 14 | Data received | The number of packets received by the node from the CH |
| 15 | Data sent to BS | The number of packets sent to the BS |
| 16 | Distance CH to BS | Distance between CH and BS |
| 17 | Send Code | The cluster sending code |
| 18 | Consumed energy | The current energy for the node in the current round |
| 19 | Attack Type | Type of the node |