Received: 1 July 2025 | Revised: 13 November 2025 | Accepted: 10 December 2025 | Published online: 27 January 2026

RESEARCH ARTICLE

Artificial Intelligence and Applications
2026, Vol. 00(00) 1–9
DOI: 10.47852/bonviewAIA62026653



BON VIEW PUBLISHING

Integrated Deep Neural Networking Approach with Long Short-Term Memory (LSTM) for Bottleneck Detection in IoT Devices

Pradip M. Paithane[1,*] and Ashwini S. Gavali[2]

[1] *Information Technology Department, Vidya Pratishthan's Kamal Nayan Bajaj Institute of Engineering & Technology, India*

[2] *CSMSS College of Engineering, Dr. Babasaheb Ambedkar Technological University, India*

Abstract: This review examines how machine learning (ML) is being used to spot bottlenecks in systems such as cloud computing, fog environments, and the Internet of Things (IoT). It covers some of the more recent developments, especially one standout technique: an amalgam deep learning model that combines DNN and LSTM algorithms to detect IoT botnet attacks. What's striking is that this method manages to correctly identify 99.98% of even the most complex attacks, and it does so in just 0.022 milliseconds—a pretty solid performance. The paper also highlights the role of convolutional autoencoders, which have shown promise in detecting suspicious activity across IoT networks, hitting an impressive accuracy rate of 99.88%. Beyond that, it dives into some innovative approaches such as software-defined networking (SDN) and other ML techniques that are specifically geared toward managing and reducing botnet-related risks. There's also discussion around newer feature selection methods and hybrid deep learning strategies that aim to boost memory efficiency while avoiding problems such as underfitting or overfitting. Overall, the review brings together a wide range of insights from recent research and points out practical tools and frameworks that could help tackle current and future cybersecurity issues across cloud, fog, and IoT systems. This approach is applied to 11 different IoT attacks with 20%, 50%, and 100% of the selected features. In this approach, an accuracy of 99.76% is achieved with 50% of the selected features, while an accuracy of 99.86% is obtained with 20% of the selected features.

Keywords: bottleneck, convolutional neural network, deep neural network, IoT devices, long short-term memory (LSTM), machine learning

1. Introduction

In today's rapidly growing world of the Internet of Things (IoT), the rising number of linked devices is driving innovation and improving efficiency across various sectors. However, this growing interconnectivity comes with increased vulnerability. One major challenge facing IoT infrastructure is the risk of security breaches, particularly in the form of system bottlenecks. This survey paper focuses on that very issue—exploring how a hybrid deep learning approach can help detect and prevent these bottlenecks [1]. By diving into this advanced technique, the paper aims to explain both how it works and why it's becoming essential in making IoT systems more secure and resilient. The importance of cybersecurity has surged in recent years, driven by the explosion of IoT devices, rapid growth of computer networks, and widespread use of digital applications in both personal and industrial spaces. Alongside this growth, the threat from malicious software—commonly known as malware—is increasing alarmingly [2]. The AV-TEST Institute reports around 350,000 novel malware threats are identified every period. These can form botnets, which allow cybercriminals to hijack countless devices and use their combined power to disrupt essential services, including those run by businesses and governments. In fact, the number of botnets discovered rose by 24% in just the first quarter of 2021. Detecting them, however, is not straightforward. Botnet attacks are not only global but also highly adaptive. With attackers constantly evolving their strategies, the task of identifying and stopping these threats remains a major hurdle—even as researchers make significant progress in detection technologies. While traditional security measures such as firewalls and encryption still play a vital role, "intrusion detection systems" (IDSs) need proven to be more operative at recognizing external threats to networked systems [3]. The hybrid deep learning method highlighted in this paper takes a more advanced route. By combining different AI models, it analyzes network performance issues—such as bottlenecks—from multiple perspectives. This makes it especially useful in strengthening IoT systems, where speed, accuracy, and adaptability are crucial. Ultimately, this paper aims to break down the working principles and real-world applications of this innovative method. The goal is to contribute meaningfully to the research community and help security professionals, developers, and decision-makers better protect increasingly complex IoT environments [4]. Feature extraction is a crucial step before learning, or training, a machine learning model. These characteristics lower the complexity of the data, improve the accuracy of ML models, and serve as discriminators for learning and inference. Flow-based features are the most frequently used in bot detection. These characteristics do not, however, fully capture the communication patterns that may reveal other facets of malevolent hosts. Furthermore, there may be a significant processing cost associated with flow-level models, which can be avoided by modifying behavioral features, such as packet structure.

When it comes to SDN-based fog computing, several key features stand out: the ability to manage secure connections for vast numbers of devices, provide real-time insights with minimal delay, and enable flexible, on-the-fly policy changes thanks to its programmable structure [5]. These qualities—programmability, adaptability, and scalability—make SDN a strong match for the needs of fog systems. To address growing security concerns, recent work has proposed using deep learning to detect botnet

***Corresponding author:** Pradip M. Paithane, Information Technology Department, Vidya Pratishthan's Kamal Nayan Bajaj Institute of Engineering & Technology, India. Email: pradip.paithane@vpkbiet.org

© The Author(s) 2026. Published by BON VIEW PUBLISHING PTE. LTD. This is an open access article under the CC BY License (https://creativecommons.org/licenses/by/4.0/).

attacks [6]. Combining SDN's architecture with a hybrid deep learning detection strategy has shown promise in improving detection accuracy and overall system performance. Limiting security risks from IoT applications may be possible with the implementation of IDS within the network. Through continuous monitoring of incoming and outgoing Internet data gathered by Internet of Things devices, the intrusion detection system detects any indications of cyberattacks. There are two types of threat identification methods: anomaly based and signature based. If recorded events satisfy the criteria obtained from known assaults, the signature-based intrusion detection system identifies a danger. In contrast, an anomaly-based approach uses the normal behavior of the state to create a simulation [6]. This method might then be used to detect assault by comparing observed and learnt behavior. These approaches' main drawback is that they would only address a small variety of processing behaviors, which is insufficient for multicomponent IoT networks. Additionally, different kinds of devices could cause diverse network behaviors, which would make attack detection less accurate. As demonstrated by the widespread use of deep learning across many industries, deep learning-based IDS may be able to overcome current obstacles and improve detection and prevention accuracy. In fact, when it comes to learning intricate and nonlinear network traffic features, the deep learning model performs better than a number of other supervised learning models. Updating the labeled datasets is not feasible because of the labor-intensive and time-consuming nature of the labeling process. Furthermore, despite their significance in improving IDS detection accuracy, high-quality datasets for training IDSs are hard to find. Privacy concerns are the main cause of this shortfall. To solve the aforementioned problems, numerous deep learning algorithms have been created. In contrast to signature-based approaches, data mining techniques aim to develop a data-driven model for infection diagnosis based on predetermined characteristics. Steady and transient strategies are the two most used ways to extract features from malware. The associated elements, such as byte sequence or string attributes, can sometimes be extracted from executable code via source code, and the static analysis was carried out in a non-runtime environment. Malware must run in a controlled environment in order to track its activities and identify hostile activity in order to do a performance simulation. Using static or dynamic evaluations, subject matter experts could create useful characteristics for a malware region identification application [7].

2. Literature Review

Malware attacks continue to be one of the biggest challenges in today's digital security landscape. As threats grow more sophisticated, the need for reliable tools to detect and stop them becomes even more urgent. One promising direction researchers are exploring involves the use of machine learning algorithms, which seem to offer solid results so far. Here's an interesting example: some approaches convert malware code into image-like representations. These are then fed into CNNs to find hidden patterns and classify the threats [8]. Pretty clever, right? But there's a catch—it can mess with the malware's natural structure, which is linear or one-dimensional. To work around that, researchers have started digging into the actual bit and byte patterns of the code. Instead of turning it into images, they've developed 1D CNNs that preserve the structure and still get accurate results. Now, zooming out a bit—let's talk about the Internet of Things (IoT). This field adds another layer of complexity to cybersecurity. Think of it as a massive web of devices—your smart fridge, industrial sensors, and everything in between—all chatting away via the Internet. Sounds futuristic, but it also creates new vulnerabilities. These devices often run on minimal hardware, and the networks they rely on aren't always secure. To make these systems safer, researchers have proposed something pretty cool: letting each IoT device generate its own security credentials rather than relying on a central authority [9]. This decentralized approach means devices can verify and communicate with each other directly. It's a practical move, especially in environments where devices operate on their own without a central hub. Of course, as with anything connected to the Internet, new risks pop up. Attacks such as malware infections or unauthorized downloads are no longer rare—they're happening all the time. These kinds of breaches can be disastrous, leading to stolen data, financial setbacks, and reputation damage. To tackle such threats more effectively, researchers are experimenting with bot detection techniques that go beyond the basics [10]. While traditional tools analyze flow-level data (such as packet counts), they can be a bit slow and may not fully grasp how devices are talking to each other. That's where a more intuitive method—graph-based analysis—comes into play. It models the relationships between devices such as a social network. This makes it easier to spot suspicious behavior. One breakthrough in this space comes from Asmae Bentaleb and her team, who designed a hybrid model called AE-CNN [1]. It combines autoencoders (AEs) with CNNs to detect anomalies in IoT networks. The autoencoder compresses and reconstructs data to highlight anything unusual, while the CNN digs into the patterns to pull out important details [11]. Their setup includes several convolution and resampling layers that help reduce and rebuild input data—essentially helping the model notice what doesn't fit. To make sure it works well, they fine-tuned the settings through hyperparameter optimization. The results are pretty impressive. On the IoT-23 dataset, their model hit an accuracy of 99.98%. Other performance markers such as precision, recall, and F1-score—tested through a tenfold cross-validation—back up its reliability for spotting threats in IoT environments.

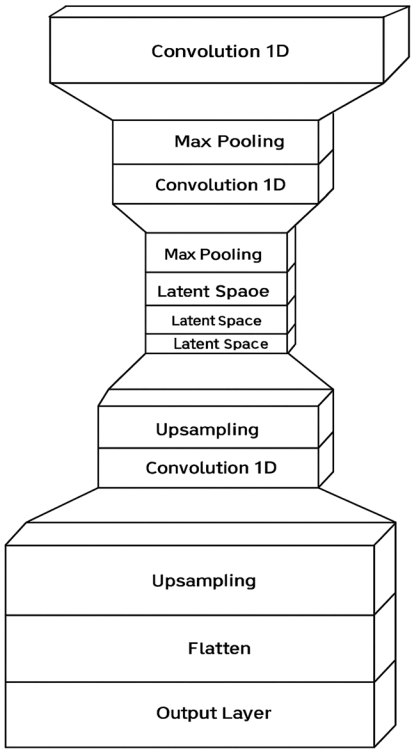
Table 1 presents a detailed literature survey of research published between 2021 and 2025, including the datasets used.

Table 1
Detailed review of recently published research articles on hybrid approaches for detecting bottleneck attacks in IoT devices

Sr. no.	Paper	Year	Approach	Dataset	Accuracy
1	"An optimized LSTM-based deep learning model for anomaly network intrusion detection" [22]	2025	Optimized LSTM-based deep learning	BoT-IoT	0.8588
2	"A New Hybrid Approach using Deep Learning in handling IoT Attack" [1]	2024	AE-CNN	IoT23	0.9988
3	"Reliable Machine Learning Model for IIoT Botnet Detection" [2]	2023	IHHO-NN	N-BaIoT	0.9807
4	"A Hybrid Deep Learning Approach for Bottleneck Detection in IoT" [3]	2022	DNN-LSTM	N_BaIoT	0.9998
5	"Hybrid Deep Learning for Botnet Attack Detection in the Internet-of-Things Networks" [4]	2021	LAE-BLSTM	BoT-IoT	0.9949
6	"A Hybrid DL-Based Detection Mechanism for Cyber Threats in Secure Networks" [5]	2021	DNN-LSTM	N_BaIoT 2018	0.9996
7	"Botnet Detection Approach Using Graph-Based Machine Learning" [6]	2021	Graph-based Ml	CTU-13	0.9975

In Figure 1, Fraidoon Sattari and colleagues [3], along with Sirajuddin Qureshi's team [5], put forward an interesting hybrid DL for spotting botnet attacks in IoT grids. What sets their tactic apart is the amalgamation of deep neural networks (DNN) and long short-term memory (LSTM) models—two powerful tools in modern AI. Their system performed remarkably well, managing to detect 99.98% of complex and diverse botnet threats, all while keeping processing time to just 0.022 milliseconds, which is seriously efficient. They tested the model using the N-BaIoT 2018 dataset, which covers a broad spectrum of IoT behaviors and threat types—making it a solid benchmark for practical circumstances. Standard assessment metrics such as precision, recall, F1-score, and accuracy, along with tenfold cross-validation, helped confirm that the model holds up well under scrutiny. All signs point to this hybrid approach being a strong contender for addressing one of the major security concerns in IoT: detecting botnet attacks effectively and quickly [11, 12].

Figure 1
Convolutional auto-encoders



In Figure 2, Fatma Taher and her team [2] took a different route in tackling botnet detection and came up with something just as impressive: the IHHO-NN model. This approach stands out for its smart use of feature selection and optimization strategies aimed at improving detection accuracy. What they did was combine hybrid filter-wrapper methods, clustering, and the Grasshopper Optimization Algorithm (GOA) to narrow down the most important features—essentially trimming the data down to just what matters most. But they didn't stop there. The model also uses the Improved Harris Hawks Optimization (IHHO) algorithm to fine-tune neural network hyperparameters. It does this with a couple of clever tricks, such as chaotic map-based initialization and an improved way of managing escape energy, which helps strike a better balance between exploring new solutions and focusing on the best ones [13, 14]. When they tested the model on the "N-BaIoT" dataset, it pulled off an accuracy rate of 99.98%—a pretty remarkable achievement, especially for multiclass botnet attack detection. Standard metrics such as precision, recall, F1-score, and overall accuracy were all used for validation, and the results clearly show that this method is highly capable of spotting even complex intrusion patterns in IoT environments [15].

In Figure 3, IHHO-NN model is discussed in detail.

In Figure 4, Segun I. Popoola and his team [4] proposed a fresh take on botnet recognition in IoT grids with a hybrid DL called LAE-BLSTM. It's an interesting combination—on one side, you have the Long Short-Term Memory Autoencoder (LAE) working to shrink feature dimensionality, which not only boosts computational efficiency but also keeps important network details intact during the process [16]. Then there's the Bidirectional Long Short-Term Memory (BLSTM) component, which handles the grouping of grid traffic. What's especially useful about BLSTM is its resilience—it deals pretty well with challenges such as underfitting and overfitting, and it adapts across different types of network traffic. The prototypical was verified thoroughly using the BoT-IoT dataset, and it performed strongly in both binary and multiclass classification settings [17]. To see how it handled different optimization strategies, the researchers tried a bunch of popular algorithms available in Keras, such as "Adam, SGD, RMSprop, Adadelta, Adagrad, Adamax, Nadam, and FTRL" [18]. They didn't just run them for comparison's sake—they really explored how each one affected the performance of the LAE model. Interestingly, their experiments showed that the combination not only maintained efficient feature reduction but also reached a solid accuracy rate of 99.49% for detecting botnet attacks in IoT systems [19, 20]. All in all, it's a well-rounded approach that seems both efficient and adaptable.

In Figure 5, Afnan Alharbi and Khalid Alsubhi [6] took on the challenge of detecting botnets by developing a smart, graph-based identification system. Their approach focuses on recognizing different types of botnets, even those with varied and unpredictable behavior patterns. What makes their method stand out is the way they engineered the features—it's not just basic preprocessing. They used five different evaluation metrics drawn from information theory, correlation, and consistency analysis to fine-tune the input data and improve how well the machine learning algorithms perform. To train and validate the model, they relied on two widely used datasets—CTU-13 and IoT-23—which helped ensure the system was tested on a good mix of botnet behaviors. The model came out with an accuracy of 97.95%, which is quite solid. Standard performance checks such as precision, recall, F1-score, and accuracy, backed by tenfold cross-validation, supported the method's overall effectiveness [21, 22]. When compared with more advanced flow-based and graph-based detection methods, their graph-based system didn't just hold its own—it actually showed better precision and stayed competitive in terms of accuracy. One of the big takeaways from this study is how effective graph-based features can be when it comes to picking up on botnet activity. The results really highlight their potential for making detection systems smarter and more adaptive [23].

3. Research Methodology

3.1. Data preprocessing

The main role of data preprocessing is to clean and transform the dataset for training and testing purposes. Through this process, the proposed model has improved accuracy by enhancing data quality, reducing noise, and addressing missing values and class imbalanced values. Different attacks are available in the said dataset, so augmentation is required; the following steps are involved in performing preprocessing on the dataset. By improving data quality, lowering noise, and resolving problems such as missing or unbalanced data, preprocessing datasets for IDS classification enhances model performance. Additionally, by strengthening the model's resistance to various data distributions and unidentified attack types, it increases the model's generalizability.

Figure 2
DNN-LSTM architecture

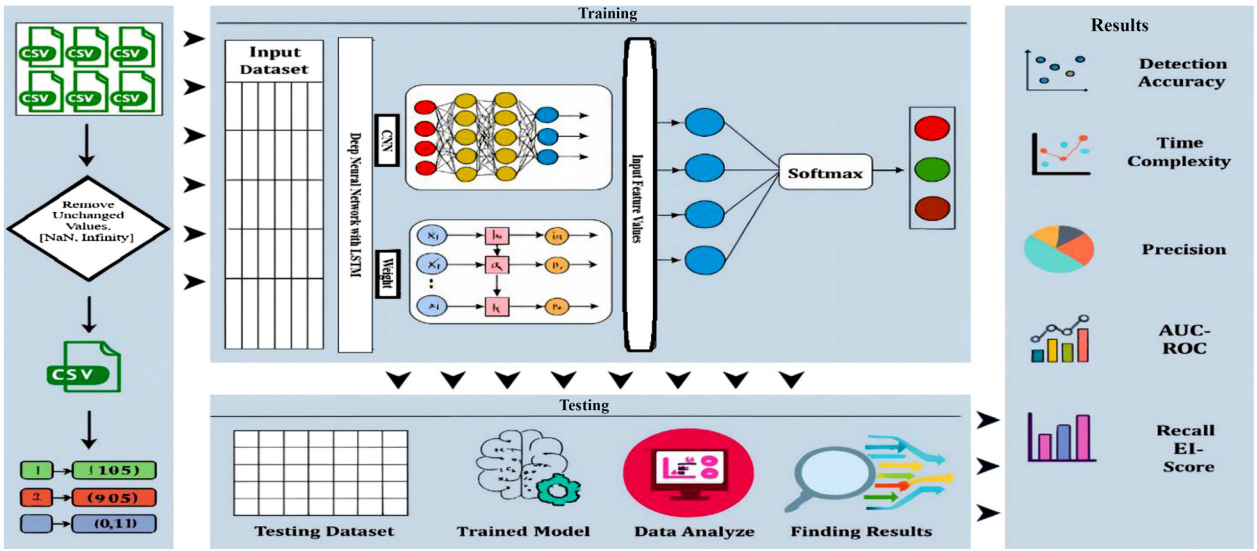


Figure 3
IHHO-NN

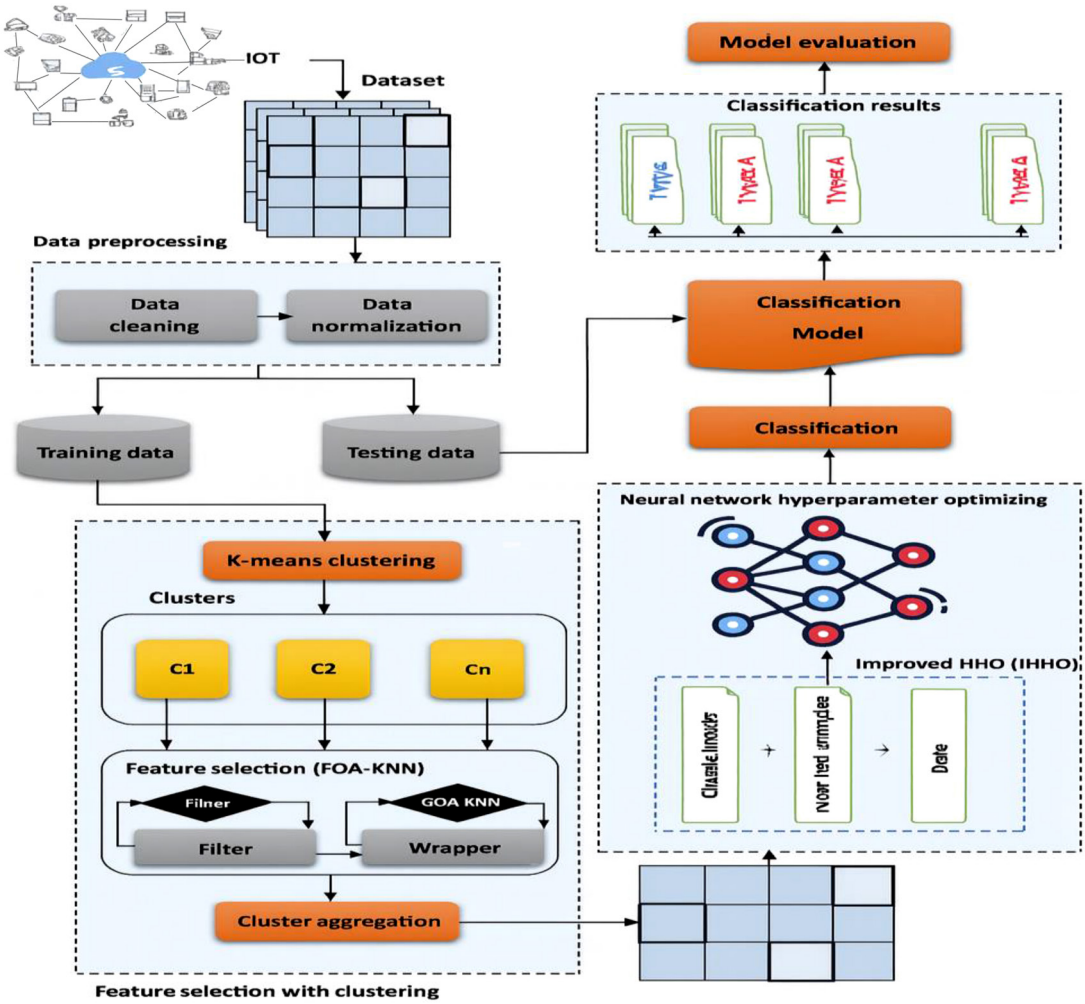


Figure 4
LAE-BLSTM

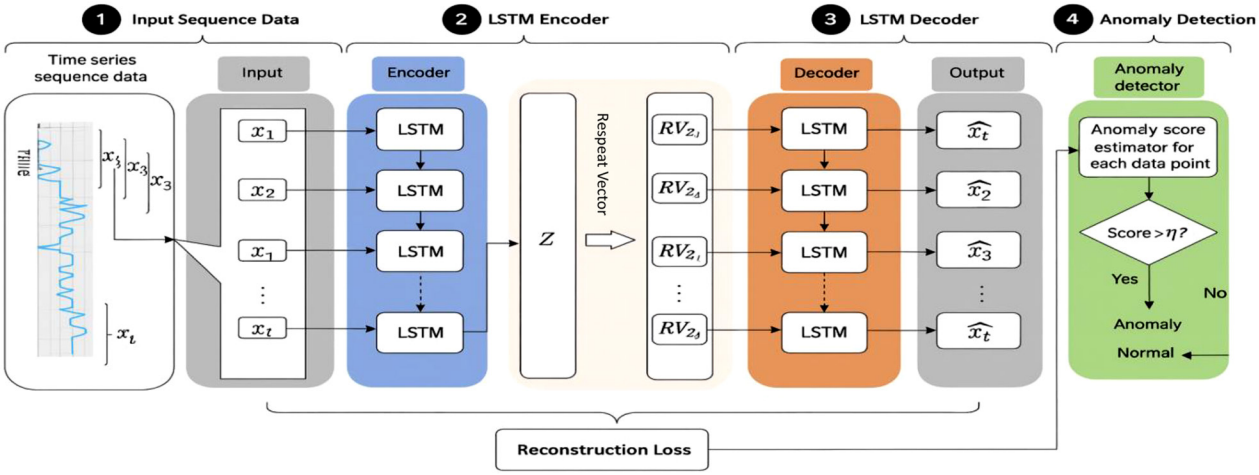
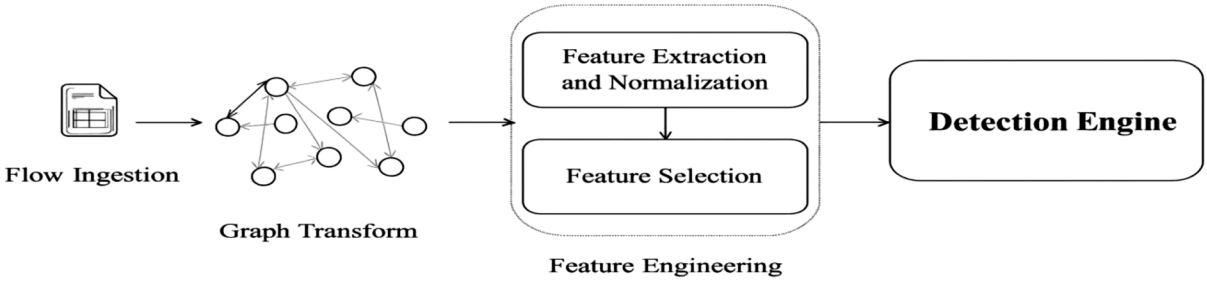


Figure 5
Graph-based machine learning model



Data cleaning: Duplicate records and missing values are present in the dataset. Duplicate records are eliminated. Imputing values and eliminating incomplete records are two ways for addressing missing values.

Data digitization: Character and numeric valued attributes are combined in the dataset. The dataset's character-valued properties are transformed into numerical values.

Data normalization: The dataset is normalized within the range of [0–1] in order to increase classification accuracy. Features with a high value range are linearly scaled to the range [0.0 to 1.0] using Equation 1. This scaling process helps to standardize and normalize the feature values for consistency in the dataset [24].

$$f = \frac{f - \min}{\max - \min} \tag{1}$$

Dataset:

1) IoT-23 is a new dataset of network traffic from Internet of Things (IoT) devices. It includes 20 clips of malware operating on IoT devices and 3 catches of traffic from benign IoT devices. It was first published in January 2020 and included photos from 2018 to 2019. This IoT network traffic was captured at the Stratosphere Laboratory, AIC group, FEL, CTU University, Czech Republic. Its goal is to give academics a large dataset of real, labeled IoT malware infections and benign traffic so they may develop machine learning techniques. This dataset and associated study were funded by Avast Software, Prague. The IoT-23 dataset consists of 23 captures (called scenarios) of different types of IoT network traffic. These scenarios are divided into 20 network captures (pcap files) from infected IoT devices (which will include the name of the malware sample executed on each scenario) and 3 network captures of actual IoT device network traffic (which include the names of the devices where the traffic was captured). For every harmful scenario, we ran a specific malware sample on a Raspberry Pi that performed a variety of functions and used numerous protocols.
2) Researchers at Ben-Gurion University in Israel created N-BaIoT, which focuses on botnet attack detection for Internet of Things (IoT) devices. Nine commercial IoT devices (including cameras and smart outlets) that were infected with BASHLITE and Mirai botnet malware are included in the collection. Each of the collection's 7,062,606 entries has 115 properties that characterize various statistical aspects of network traffic. Ten classes comprise the attack types, which include various DDoS and remote access attacks [25].

3.2. Autoencoders

Autoencoders (AE) are unsupervised neural networks used for dimensionality reduction and feature learning by reconstructing input data:

1) Encoder function ("f"): The encoder function "f" learns and represents input "x" as a code, defined by

$$f(x) = ae(wx + b) \tag{2}$$

2) Decoder function ("g"): The decoder function "g" reconstructs data from the encoded representation, expressed as

$$\hat{x} = g(z) = g(f(x)) = ad(w_0 \bullet f(x) + b_0) \tag{3}$$

3) Hidden layer ("h"): The hidden layer "h" serves as a code describing the input.

With the help of autoencoders' function, Autoencoders are a family of unsupervised deep learning models that use an encoder-decoder structure to recreate input data in order to develop effective data representations. While the decoder works to recreate the original input from this compact representation, the encoder compresses the input into a lower-dimensional latent space. Autoencoders are very useful for detecting IoT abnormalities or network intrusions because they can simulate typical network traffic flow and spot deviations as possible anomalies. The reconstruction error for typical patterns stays low when trained only on benign data, but aberrant or malicious traffic results in far larger reconstruction errors, which are a symptom of infiltration or penetration. Autoencoders can have some drawbacks, though. The quality and variety of training data have a significant impact on their effectiveness, and if undetected patterns significantly deviate from the training distribution, they may have trouble identifying zero-day assaults. Furthermore, it is challenging to understand why particular samples are marked as anomalies because of their intrinsic lack of explainability.

3.3. Deep neural network (DNN)

A deep neural network (DNN) consists of input, hidden, and output layers, excelling at managing unstructured data. Deep learning, akin to artificial consciousness, allows automatic identification of crucial features without explicit feature selection [26].

The computational complexity of a DNN is expressed as:

$$m_i = f\left(\sum_{i=1}^{S} y_i + x_i + v\right) \tag{4}$$

1) m_i represents the output weight associated with input i.
2) f() denotes the training function applied to the summed inputs.
3) $\sum_{i=0}^{S}$ signifies the summation across s inputs.
4) y_i stands for the input weights.
5) x_i refers to the input values.
6) v denotes the bias vector.

3.4. Deep neural network with long short-term memory (DNN-LSTM)

Long short-term memory (LSTM) is a neural network architecture designed specifically for processing sequential data. LSTM excels in learning sequential patterns by incorporating information from previous time steps.

1) **Input gate:** Determines which information to store from the input and previous cell state using the sigmoid activation function

$$a_t = \sigma(w_a[n_{t-1}, x_t] + b_a) \tag{5}$$

2) **Forget gate:** Decides which information to discard from the previous cell state using the sigmoid activation function

$$i_t = \sigma(w_i[n_{t-1}, x_t] + b_i) \tag{6}$$

3) **Candidate memory:** Proposes new values to be added to the cell state using the hyperbolic tangent activation function

$$\widehat{C}_t = \tanh(w_c[n_{t-1}, x_t] + b_c) \tag{7}$$

4) **Cell state update:** Updates the cell state by combining information from the input gate and the forget gate

$$C_t = a_t * C_{t-1} + i_t * \widehat{C}_t \tag{8}$$

5) **Output gate:** Determines which information from the cell state should be outputted using the sigmoid activation function

$$O_t = \sigma(w_0[n_{t-1}, x_t] + b_0) \tag{9}$$

6) **Output:** Generates the output of the LSTM unit by combining the cell state with the output gate activation

$$n_t = O_t * \tanh(C_t) \tag{10}$$

The study highlights convolutional autoencoders (CAE) and a hybrid deep learning system combining DNN and LSTM as the most accurate models for identifying malicious activity in IoT networks.

The convolutional autoencoder (CAE) model is built around a five-layer neural network, including a bottleneck layer that compresses data into a more manageable, lower-dimensional format. In practical testing, this design turned out to be quite effective—it reached a detection accuracy of 99.88% across several types of attacks. With IoT devices multiplying rapidly and cyber threats growing more advanced, the study makes a strong case for developing fast, dependable intrusion detection systems [27]. The researchers used the IoT-23 dataset to evaluate the CAE model, and the results were promising. The system was able to identify a range of network-based threats accurately in typical IoT environments. That said, the authors point out that for the model to be truly reliable, it needs to be tested more widely—across different networks and datasets—to see how well it holds up under varied conditions [28].

Beyond CAE, the study also introduces a hybrid deep learning method that brings together deep neural networks (DNN) and long short-term memory (LSTM) models. This combo proved especially good at spotting tricky, multivariant attack patterns that often appear in IoT setups [29]. Moreover, it delivers those results fast—with an average processing time of just 0.022 milliseconds. That kind of speed, paired with a 99.98% detection rate, puts it ahead of many older techniques when it comes to detecting sophisticated botnet attacks.

Given how persistent and disruptive botnets have become in the IoT world, this solution feels like a timely and important step forward [30]. The study takes advantage of LSTM's strength in processing sequential data, applying both DNN and LSTM models to the N-BaIoT dataset, which features a mix of benign and malicious activity from various botnet families [31, 32]. They didn't just rely on basic accuracy either—the hybrid model was evaluated using both standard and extended metrics. Across the board, it showed strong potential in addressing some of today's most pressing IoT security challenges.

3.5. Contribution

1) Bottleneck recognition prototype is proposed in this study to identify the bottleneck attacks and protect against cyber threats using machine learning (ML) algorithms in the context of cloud, fog computing, and the IoT.
2) An AE-CNN-based model uses the "IoT-23" dataset, which contains 23 scenarios of IoT network traffic, including 20 "pcap" files from infected IoT devices. Additionally, a DNN-LSTM hybrid model leverages the N-BaIoT dataset, comprising 117 properties that capture both legitimate and malicious IoT activity traces.
3) The convolutional autoencoder (AE-CNN) model uses a five-layer neural network architecture, including a bottleneck layer and encoder and decoder layers, achieving a remarkable accuracy rate of 99.88%. Additionally, a hybrid deep learning system combining DNN and LSTM excels in identifying complex multi-attacks and is adept at recognizing sequence data, making it suitable for identifying botnet assaults on IoT devices, achieving a remarkable accuracy rate of 99.98%.

Evaluation criteria including accuracy, precision, recall, specificity, and F1-score must be used to assess the suggested methodology.

Figure 6 illustrates the accuracy results of the graph-based machine learning model.

Figure 6
Graph-based machine learning model

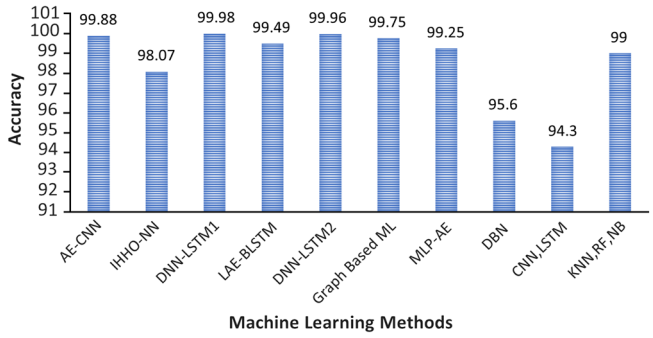


Table 2 presents a comparison of all models using various evaluation parameters.

Table 2
Comparison of results with existing approaches

Method	Accuracy	Precision	Recall	F1-score	Detection time (ms)
AE-CNN	99.88	99.20	99.12	93.15	1.2
IHHO-NN	98.07	98.50	98.92	98.44	0.56
DNN-LSTM1	99.98	99.76	99.64	99.68	0.22
LAE-BLSTM	99.49	99.12	99.46	99.33	2.65
DNN-LSTM2	99.96	99.49	99.54	99.64	0.26
Graph-Based ML	99.75	99.63	99.20	99.45	3.45
MLP-AE	99.25	98.48	96.25	99.45	4.55
DBN	95.60	98.27	92.82	92.82	8.94
CNN, LSTM	94.30	93.48	93.67	93.58	1.66
KNN, RF, NB	99.00	86.65	99.00	99.00	21.22

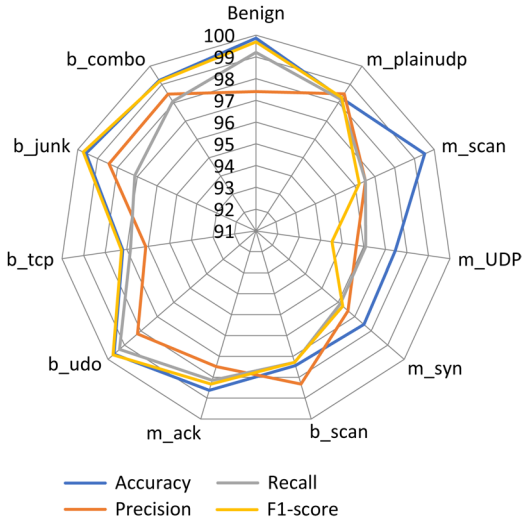
Table 3 presents the results of various attacks in detail.

Table 3
Types of attacks for DNN-LSTM1 results without hyperparameter optimization and feature selection

Types of attacks	Accuracy	Precision	Recall	F1-score
Benign	99.86	97.4	99.2	99.69
m_plainudp	98.26	98.49	98.19	98.28
m_scan	99.52	96.52	96.52	96.21
m_UDP	97.46	95.66	96.09	94.52
m_syn	97.55	96.59	96.11	96.27
b_scan	97.44	98.32	97.29	97.28
m_ack	98.63	97.49	98.15	98.33
b_udo	99.63	98.21	99.30	99.69
b_tcp	97.18	96.12	96.83	97.24
b_junk	99.59	98.43	97.12	99.72
b_combo	99.22	98.48	98.08	99.19
Average	98.57	97.43	97.53	97.86

In Figure 7, results of various attacks are shown through various evaluation parameters.

Figure 7
Results of the types of attacks using DNN-LSTM1



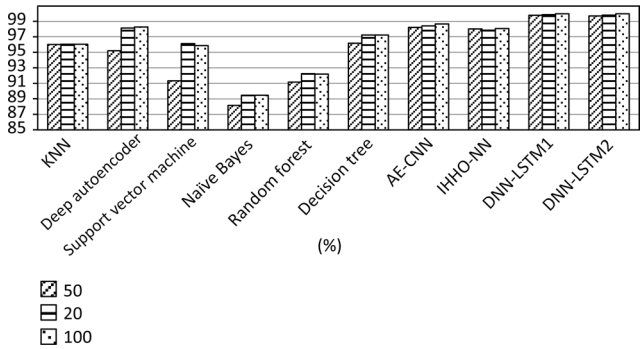
In Table 4, the comparison of DNN-LSTM1 and DNN-LSTM2 is discussed with the accuracy parameter.

Table 4
Comparison of DNN-LMST1 and DNN-LSTM2 classification accuracy using different percentages of feature selections

Classifier model	50%	20%	100%
KNN	96.01	96.03	96.03
Deep autoencoder	95.21	98.13	98.27
Support vector machine	91.33	96.12	95.88
Naïve Bayes	88.15	89.45	89.45
Random forest	91.16	92.24	92.19
Decision tree	96.17	97.23	97.23
AE-CNN	98.20	98.40	98.65
IHHO-NN	98.03	97.84	98.07
DNN-LSTM1	99.76	99.86	99.98
DNN-LSTM2	99.68	99.79	99.96

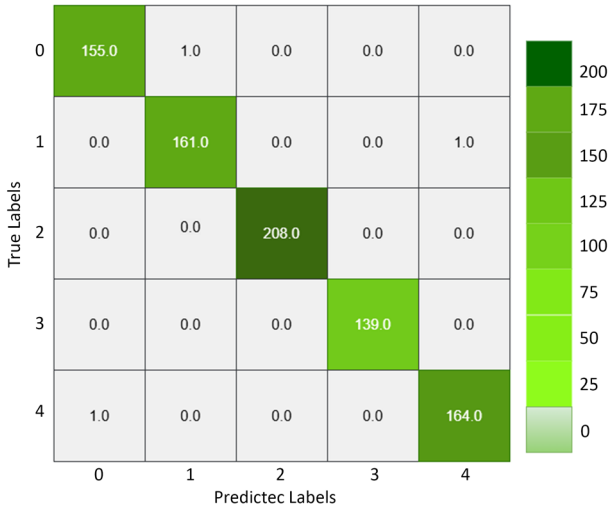
In Figure 8, the DNN and LSTM results are compared using various machine learning approaches.

Figure 8
Comparison of the classification accuracy of DNN-LMST1 and DNN-LSTM2 with varying feature selection percentages



In Figure 9, confusion matrix is shown to represent the values. These algorithms can detect even the smallest variations in traffic patterns, providing early warnings of potential intrusions long before they become widespread attacks [33]. Deep learning models are particularly adept at identifying complex intrusions such as polymorphic malware, which alters its code to evade detection using traditional methods. These models may learn and generalize from the structural features of the traffic and the payload, significantly raising detection rates regardless of how the malware evolves. By precisely identifying hostile traffic and lowering false alarms, machine learning improves threat categorization accuracy.

Figure 9
Confusion matrix of proposed approach



4. Conclusion

This research investigated several hybrid deep learning approaches for detecting IoT-based botnet attacks and network bottlenecks, focusing on the integration of DNN, LSTM, and AE. The findings highlight that both AE-CNN and DNN-LSTM models delivered exceptional detection accuracy—99.88% and 99.98%, respectively—demonstrating the ability of hybrid architectures to effectively learn complex attack behaviors and temporal dependencies in IoT network traffic. Using the "IoT-23" and "N-BaIoT" datasets, the proposed models were able to accurately identify diverse IoT attack scenarios with impressive speed and stability across key performance metrics such as accuracy, precision, recall, and F1-score. Among the tested models, the DNN-LSTM hybrid framework proved particularly strong, owing to its capability to analyze sequential data patterns and adapt dynamically to fluctuating network conditions. Looking ahead, future work should aim to extend the evaluation to real-time IoT environments, implement automated hyperparameter optimization, and strengthen the models against adversarial attacks. Enhancing interpretability and incorporating domain-specific intelligence can further increase trust and reliability. Additionally, developing lightweight and energy-efficient architectures will be critical for utilization on resource-constrained IoT edge plans. In summary, the proposed framework represents a promising step toward intelligent, adaptive, and high-precision intrusion detection systems that can effectively combat the ever-evolving cybersecurity challenges in modern IoT ecosystems.

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

The data that support the findings of this study are openly available in N-BaIoT at https://dx.doi.org/10.21227/y9de-qj71.

Author Contribution Statement

Pradip M. Paithane: Conceptualization, Methodology, Software, Validation, Formal analysis, Writing – original draft, Writing – review & editing, Visualization, Supervision, Project administration. **Ashwini S. Gavali:** Methodology, Validation, Formal analysis, Investigation, Resources, Data Curation, Writing – review & editing, Visualization, Supervision, Project administration.

References

[1] Bentaleb, A., Remmach, C., & Abouchabaka, J. (2023). A New Hybrid Approach using Deep Learning in handling IoT Attack. In *International Conference on Intelligent Systems: Theories and Applications*, 1–7. https://doi.org/10.1109/SITA60746.2023.10373696

[2] Taher, F., Abdel-Salam, M., Elhoseny, M., & El-Hasnony, I. M. (2024). Reliable machine learning model for IIoT botnet detection. *IEEE Access*, *11*, 49319–49336. https://doi.org/10.1109/ACCESS.2023.3253432

[3] Sattari, F., Farooqi, A. H., Qadir, Z., Raza, B., Nazari, H., & Almutiry, M. (2022). A hybrid deep learning approach for bottleneck detection in IoT. *IEEE Access*, *10*, 77039–77053. https://doi.org/10.1109/ACCESS.2022.3188635

[4] Popoola, S. I., Adebisi, B., Hammoudeh, M., Gui, G., & Gacanin, H. (2020). Hybrid deep learning for botnet attack detection in the internet-of-things networks. *IEEE Internet of Things Journal*, *8*(6), 4944–4956. https://doi.org/10.1109/JIOT.2020.3034156

[5] Qureshi, S., He, J., Tunio, S., Zhu, N., Akhtar, F., Ullah, F.,…& Wajahat, A. (2021). A hybrid DL-based detection mechanism for cyber threats in secure networks. *IEEE Access*, *9*, 73938–73947. https://doi.org/10.1109/ACCESS.2021.3081069

[6] Alharbi, A., & Alsubhi, K. (2021). Botnet detection approach using graph-based machine learning. *IEEE Access*, *9*, 99166–99180. https://doi.org/10.1109/ACCESS.2021.3094183

[7] Wagh, S. J., Paithane, P. M., & Patil, S. N. (2021). Applications of fuzzy logic in assessment of groundwater quality index from Jafrabad Taluka of Marathawada region of Maharashtra State: A GIS based approach. In *International Conference on Hybrid Intelligent Systems*, 354–364. https://doi.org/10.1007/978-3-030-96305-7_33

[8] Nagaraju, R., Pentang, J. T., Abdufattokhov, S., CosioBorda, R. F., Mageswari, N., & Uganya, G. (2022). Attack prevention in IoT through hybrid optimization mechanism and deep learning framework. Measurement: *Sensors*, *24*, 100431. https://doi.org/10.1016/j.measen.2022.100431

[9] Rahmantyo, D. T., Erfianto, B., & Satrya, G. B. (2021). Deep residual CNN for preventing botnet attacks on the internet of things. In *International Conference of Computer and Informatics Engineering*, 462–466. https://doi.org/10.1109/IC2IE53219.2021.9649314

[10] Paithane, P. M. (2023). Random forest algorithm use for crop recommendation. *ITEGAM-JETIA*, *9*(43), 34–41. https://doi.org/10.5935/jetia.v9i43.906

[11] Nguyen, T. N., Ngo, Q. D., Nguyen, H. T., & Nguyen, G. L. (2022). An advanced computing approach for IoT-botnet detection in industrial Internet of Things. *IEEE Transactions on Industrial Informatics*, *18*(11), 8298–8306. https://doi.org/10.1109/TII.2022.3152814

[12] Hussain, F., Abbas, S. G., Pires, I. M., Tanveer, S., Fayyaz, U. U., Garcia, N. M., ... & Shahzad, F. (2021). A two-fold machine learning approach to prevent and detect IoT botnet attacks. *IEEE Access*, *9*, 163412–163430. https://doi.org/10.1109/ACCESS.2021.3131014

[13] Vu, L., Nguyen, Q. U., Nguyen, D. N., Hoang, D. T., & Dutkiewicz, E. (2020). Deep transfer learning for IoT attack detection. *IEEE Access*, *8*, 107335–107344. https://doi.org/10.1109/ACCESS.2020.3000476

[14] Mahajan, D. D., & Jeyasekar, A. (2025). Deep Shallow network with LSTM for detecting attacks in IoT networks and preserving privacy based on adaptive hybrid encryption algorithm. *Expert Systems with Applications*, 128050. https://doi.org/10.1016/j.eswa.2025.128050

[15] Prokofiev, A. O., Smirnova, Y. S., & Surov, V. A. (2018). A method to detect Internet of Things botnets. In *IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering*, 105–108. https://doi.org/10.1109/EIConRus.2018.8317041

[16] Meidan, Y., Bohadana, M., Mathov, Y., Mirsky, Y., Shabtai, A., Breitenbacher, D., & Elovici, Y. (2018). N-BaIoT—network-based detection of IoT botnet attacks using deep autoencoders. *IEEE Pervasive Computing*, *17*(3), 12–22. https://doi.org/10.1109/MPRV.2018.03367731

[17] Mahalle, P. N., Talware, R. S., Patil, G. C., Sakhare, S. R., Dandawate, Y. H., & Futane, P. R. (2022). *Artificial intelligence in information and communication technologies, healthcare and education: A roadmap ahead*. USA: CRC Press.

[18] Malik, J., Akhunzada, A., Bibi, I., Talha, M., Jan, M. A., & Usman, M. (2020). Security-aware data-driven intelligent transportation systems. *IEEE Sensors Journal*, *21*(14), 15859–15866. https://doi.org/10.1109/JSEN.2020.3012046

[19] Ning, Z., Hu, X., Chen, Z., Zhou, M., Hu, B., Cheng, J., & Obaidat, M. S. (2017). A cooperative quality-aware service access system for social Internet of vehicles. *IEEE Internet of Things Journal*, *5*(4), 2506–2517. https://doi.org/10.1109/JIOT.2017.2764259

[20] Arnob, A. K. B., Mridha, M. F., Safran, M., Amiruzzaman, M., & Islam, M. R. (2025). An enhanced LSTM approach for detecting IoT-based DDoS attacks using honeypot data. *International Journal of Computational Intelligence Systems*, *18*(1), 19. https://doi.org/10.1007/s44196-025-00741-7

[21] Wang, X., Ning, Z., Zhou, M., Hu, X., Wang, L., Zhang, Y.,…& Hu, B. (2018). Privacy-preserving content dissemination for vehicular social networks: Challenges and solutions. *IEEE Communications Surveys & Tutorials*, *21*(2), 1314–1345. https://doi.org/10.1109/COMST.2018.2882064

[22] Dash, N., Chakravarty, S., Rath, A. K., Giri, N. C., AboRas, K. M., & Gowtham, N. (2025). An optimized LSTM-based deep learning model for anomaly network intrusion detection. *Scientific Reports*, *15*(1), 1554. https://doi.org/10.1038/s41598-025-85248-z

[23] Abou Daya, A., Salahuddin, M. A., Limam, N., & Boutaba, R. (2020). BotChase: Graph-based bot detection using machine learning. *IEEE Transactions on Network and Service Management*, *17*(1), 15–29. https://doi.org/10.1109/TNSM.2020.2972405

[24] Asadi, M., Heidari, A., & Jafari Navimipour, N. (2025). A new flow-based approach for enhancing botnet detection efficiency using convolutional neural networks and long short-term memory. *Knowledge and Information Systems*, *67*(7), 6139–6170. https://doi.org/10.1007/s10115-025-02410-9

[25] Meidan, Y. (2025). *N-BaIoT* [Date Set]. https://dx.doi.org/10.21227/y9de-qj71

[26] Raheja, S., Munjal, G., Jangra, J., & Garg, R. (2021). Rule-based approach for botnet behavior analysis. In S. K. Pani, S. K. Singh, L. Garg, R. B. Pachori & X. Zhang(Eds), *Intelligent data analytics for terror threat prediction: Architectures, methodologies, techniques and applications* (pp. 161–179). Wiley. https://doi.org/10.1002/9781119711629.ch8

[27] Rosenthal, G., Kdosha, O. E., Cohen, K., Freund, A., Bartik, A., & Ron, A. (2020). ARBA: Anomaly and reputation based approach for detecting infected IoT devices. *IEEE Access*, *8*, 145751–145767. https://doi.org/10.1109/ACCESS.2020.3014619

[28] Al-Sarem, M., Saeed, F., Alkhammash, E. H., & Alghamdi, N. S. (2021). An aggregated mutual information based feature selection with machine learning methods for enhancing IoT botnet attack detection. *Sensors*, *22*(1), 185. https://doi.org/10.3390/s22010185

[29] Abu Khurma, R., Almomani, I., & Aljarah, I. (2021). IoT botnet detection using salp swarm and ant lion hybrid optimization model. *Symmetry*, *13*(8), 1377. https://doi.org/10.3390/sym13081377

[30] Shahin, M. (2022). Evaluating the fidelity and efficiency of network intrusion detection systems via deep learning, machine learning, and deep hybrid learning in industrial IoT devices. Doctoral Dissertation, University of Texas at San Antonio.

[31] Ahmad, Z., Shahid Khan, A., Wai Shiang, C., Abdullah, J., & Ahmad, F. (2021). Network intrusion detection system: A systematic study of machine learning and deep learning approaches. *Transactions on Emerging Telecommunications Technologies*, *32*(1), e4150. https://doi.org/10.1002/ett.4150

[32] Ali, M. H., Jaber, M. M., Abd, S. K., Rehman, A., Awan, M. J., Damaševičius, R., & Bahaj, S. A. (2022). Threat analysis and distributed denial of service (DDoS) attack recognition in the internet of things (IoT). *Electronics*, *11*(3), 494. https://doi.org/10.3390/electronics11030494

[33] Mohamed, N. (2025). Artificial intelligence and machine learning in cybersecurity: A deep dive into state-of-the-art techniques and future paradigms. *Knowledge and Information Systems*, *67*, 6969–7055. https://doi.org/10.1007/s10115-025-02429-y

How to Cite: Paithane, P. M., & Gavali, A. S. (2026). Integrated Deep Neural Networking Approach with Long Short-Term Memory (LSTM) for Bottleneck Detection in IoT Devices. *Artificial Intelligence and Applications*. https://doi.org/10.47852/bonviewAIA62026653