

Strengthening Cybersecurity: DDoS Attack Detection with Deep Learning and Innovative Hybrid Methods

Josías Chávez Campoverde¹, Misael Chávez Campoverde², Daniel Chávez Campoverde³, Naomi Chávez Campoverde⁴, Jorge Chávez Díaz⁵

¹Department of Systems Engineering, Faculty of Engineering, Universidad de Lima, Lima, Perú

²Faculty of Communications, Instituto de Educación Superior Toulouse-Lautrec, Lima, Perú

³Faculty of Engineering, Centennial College, Toronto, Canada

⁴Department Computing Studies & Information Systems, Faculty Commerce & Business Administration, Douglas College, Vancouver, Canada

⁵Postgraduate School, Universidad Peruana de Ciencias Aplicadas, Lima, Perú

Article Info

Article history:

Received Sept 15, 2025

Revised Dec 12, 2025

Accepted Dec 31, 2025

Keywords:

Cybersecurity

DDoS attacks

Deep learning

Hybrid neural networks

Digital infrastructure

ABSTRACT

Distributed Denial-of-Service (DDoS) attacks continue to disrupt the availability of online services, motivating the development of robust and scalable detection mechanisms. This work proposes a hybrid CNN-LSTM detection framework evaluated in a controlled, sandboxed testbed for traffic generation and monitoring. The framework is implemented under a supervised learning setting and is positioned to incorporate semi-supervised and transfer learning strategies to address label scarcity and potential distribution shift in future extensions. Using a dataset of 6,000 labeled traffic logs and an 80/10/10 train/validation/test split, the proposed model achieves 98.67% accuracy, 98.01% precision, 96.73% recall, and 97.37% F1-score, outperforming Random Forest (96.42%) and a standalone LSTM (97.10%). Overall, the hybrid design supports improved detection robustness and can serve as a practical component within layered DDoS defense strategies (e.g., filtering and elastic scaling) in operational environments.

Copyright © 2025 Institute of Advanced Engineering and Science.
All rights reserved.

Corresponding Author:

Jorge Chávez Díaz

Postgraduate School, Universidad Peruana de Ciencias Aplicadas

Av. Alonso de Molina Nro. 1611, Lima, Perú

Email: pejochav@upc.edu.pe

1. INTRODUCTION

1.1. Problems and objectives

In the context of a dynamic and constantly changing digital environment, protecting infrastructures against Distributed Denial of Service (DDoS) attacks has become a crucial priority. These attacks, capable of disrupting vital online services, represent a significant threat to the stability and reliability of digital systems. In the face of this growing concern, there is a pressing need to develop effective solutions for detecting and defending against such threats.

To address this challenge, our strategy will focus on the design and implementation of a DDoS attack detection system based on advanced deep learning techniques. A thorough evaluation of various strategies and approaches available in the detection of this type of attacks will be carried out, paying special attention to the effectiveness and efficiency of deep learning models to identify malicious traffic patterns. Through this meticulous evaluation process, we will select the best practices and methodologies for the implementation of our detection system.

Although no single measure guarantees immunity against DDoS attacks, layered mitigation strategies can reduce exposure and improve service resilience. Common approaches include traffic filtering and reputation-based blocking, the use of distributed content delivery infrastructures, and elastic scaling policies

that preserve availability under sudden workload surges. In this context, detection models that reliably discriminate baseline traffic from stress conditions can support timely activation of mitigation controls.

The objective is to develop a robust and efficient system that strengthens cybersecurity, contributing directly to the achievement of Sustainable Development Goals – (SDG) - 9: "Industry, Innovation and Infrastructure". By driving innovation in cybersecurity and improving the resilience of digital infrastructures, we aim to create a more secure and trusted digital environment to foster the sustainable development of industry and infrastructure.

1.2. Impact of DDoS attacks on the financial sphere

DDoS attacks carried out on company services or a certain group of people can cause different consequences and have a different impact depending on the problem or what the attacker is trying to achieve as a goal. DDoS attacks can have a significant financial impact on companies [1-3]. The impact includes lost revenue due to website downtime, mitigation costs such as hiring cybersecurity experts or acquiring new hardware and software, reputational damage and loss of customer confidence, and potential legal fees and fines if the attack results in a data breach [4-7]

A notable example is the DDoS attack on Dyn (Internet domain name allocation company) in 2016, which affected large companies such as Twitter and Netflix, showing how even large corporations can suffer massive financial losses. This will consequently not only affect the financial sphere, but also the operational part of the organization, where they manage online contacts and communicate with customers, could be triggered and spoil factors such as the reputation of the company or the operational part where they manage online contacts and communicate with customers [2, 8, 9].

1.3. Legal implications

The legal consequences of computer attacks are severely punished. For example, in the United States, laws such as the Computer Fraud and Abuse Act have been implemented that criminalize hacking and other cybercrimes [10]. However, globally, many countries lack comprehensive cyber legislation, making it difficult to effectively address and prosecute cyber-attacks. Significant legal loopholes exist, and existing international legal frameworks often struggle to cover the full range of cyberattacks [11].

1.4. Ping Flood Attack

One of the most well-known DDoS attacks are those that take advantage of the Internet Control Message Protocol (ICMP), as it uses echo requests and echo replies, or pings, to check the status and connectivity of a device [12-14]. It should be noted that this type of attack is also known as "Ping Flood", since the attacker saturates the bandwidth of a server or a page by means of thousands of requests to its IP address. As the attacker sends many packets to the victim, the latter will consume most of the resources it has, damaging the request processing capacity and, as a result, disabling the service to legitimate users [15, 16]. This type of attack is usually much more effective when it is a DDoS attack rather than a DoS attack. Since, when performing a DoS attack, only one attacker is involved, and the victim can easily trace the offending individual. On the other hand, when it is a DDoS attack, there are several actors involved which are collectively called Botnet, and all of them have a fake IP address and are difficult to locate. It is therefore very difficult for the victim to detect who is sending large numbers of requests to their IP address.

1.5. Botnets in DDoS attacks

A botnet is a network of malware-infected devices that are remotely controlled by an attacker, known as a "botmaster" [17-19]. These devices, which can include computers, smartphones and other Internet-connected devices, are called "bots" or "zombies". Botnets are mainly used for malicious activities, most notably DDoS (Distributed Denial of Service) attacks, where a large amount of traffic is sent to a server or network to saturate it and make it inaccessible. It targets the theft of personal and sensitive information, such as login credentials and banking data, the mass sending of unwanted emails or spam, click fraud, generating fake clicks on pay-per-click ads to obtain fraudulent revenue, and cryptocurrency mining, using the resources of infected devices without the consent of their owners [17, 18].

Botnets can use different control architectures, such as those based on IRC (Internet Relay Chat), where bots communicate through chat channels; on HTTP/HTTPS, using web protocols; or on P2P (Peer-to-Peer), where each bot communicates with other bots, eliminating the need for a central C&C server and making them difficult to dismantle. Detecting and mitigating botnets represents an ongoing challenge for cybersecurity due to their ability to evolve and employ advanced evasion techniques [20-22]. A command-and-control server (C&C server) is a computer that issues directives to digital devices that have been infected with rootkits or other types of malwares, such as ransomware.

1.6. Use of deep learning to detect DDoS attacks

Recent literature on DDoS detection shows a growing emphasis on hybrid strategies aimed at improving generalization in the face of unseen attacks and formalizing the stages of analysis (preprocessing, feature selection, and validation) with greater rigor. Along these lines, Najafimehr et al. propose a scheme that integrates unsupervised and supervised learning: first, they separate anomalous traffic by clustering with flow-based variables and then use statistical measures to label the clusters and classify them. They also train on CICIDS2017 and test on a more recent dataset (CICDDoS2019), reinforcing the argument for out-of-sample generalization [23]. In addition, Batchu and Seetha develop a generalized model that reduces the variable space—to decrease overfitting and computational cost—through hybrid feature selection and hyperparameter tuning, emphasizing that performance robustness critically depends on explicit and justified decisions at these stages [24]. From a more classical IDS perspective, Sumathi et al. report models based on C4.5, SVM, and KNN with cross-validation, which is useful as a methodological baseline when comparing modern hybrid proposals with traditional ML approaches [25].

At the same time, recent empirical evidence supports the idea that hybrid approaches with deep networks offer advantages when the goal is to capture nonlinear patterns and temporal dependencies without relying exclusively on manual variable engineering. In IEEE Access, Wei et al. present AE-MLP, where an autoencoder performs feature extraction/compression and an MLP classifies attack types; the design is justified by its ability to mitigate noise and overhead by operating with reduced representations and is evaluated on CICDDoS2019 with metrics above 98% (accuracy and F1) [26]. In Applied Sciences, Alghazzawi et al. combine variable selection via chi-square testing with CNN + BiLSTM architecture, articulating a pipeline where prior selection limits dimensionality and BiLSTM models sequential traffic dependency, strengthening the mathematical consistency of the method by linking filtering statistics with temporal deep learning [27]. Elsaedy et al. also proposes a hybrid model for smart city (IoT) scenarios, evaluated on real-life datasets with simulated attacks and highly accurate results, providing evidence of transferability to applied contexts beyond standard test beds [28].

Kadam and Sekhar propose a hybrid KNN–SVM (KSVM) scheme for DDoS detection in VANET, reporting improvements in accuracy and sensitivity compared to individual ML classifiers. This background shows that the hybridization of shallow models can increase performance; however, their ability to capture temporal dependencies and complex traffic patterns is limited, which justifies exploring hybrid architectures based on deep learning and adaptation strategies [29].

The most recent work tends to consolidate hybrid domain-oriented architectures (SDN and IoT) and to explain in greater detail the "mathematical" components of the analytical flow (imbalance management, correlational selection, model fusion). In SDN, Elubeyd and Yiltas-Kaplan integrate multiple deep learning algorithms into a hybrid approach, reporting accuracy close to 99.8% in two data sets, which reinforces the convenience of accurately describing the architecture and experimental protocol to support valid comparisons [30]. For their part, Al-zubidi et al. propose a hybrid CNN–LSTM–XGBoost system evaluated on three datasets (CICIDS-001, CIC-IDS2017, and CIC-IDS2018) and explicitly incorporate null/duplicate removal, imbalance treatment, and correlational selection, showing improvements in accuracy and reduction of overfitting through the selection of relevant features [31]. In IoT, Ain et al. integrate Autoencoder + LSTM + CNN on CICIOT2023 and, in addition to reporting performance, acknowledge limitations associated with rare attacks and imbalance, which justifies the manuscript clearly describing how these factors are controlled so as not to overestimate results [32].

Recent advances in sequential modeling highlight that systematic hyperparameter optimization and noise-robust recurrent architectures can substantially improve prediction reliability in dynamic conditions. For instance, improved Bayesian hyperparameter optimization combined with bidirectional LSTM has been shown to enhance estimation precision through principled tuning and robustness mechanisms. Likewise, anti-noise adaptive LSTM modeling introduces feedback correction strategies to mitigate noise and improve stability. Moreover, real-time coefficient correction in multi-feature electrochemical–thermal coupling models illustrates the benefit of online adjustment under changing regimes. Although these studies originate from battery modeling, the underlying methodological principles—robust sequential learning, principled hyperparameter tuning, and adaptive correction—are directly applicable to DDoS traffic streams characterized by drift and measurement noise [33].

The use of Deep Learning (DL) techniques has emerged as a promising solution for the detection of DDoS attacks [34–36]. DL methods can extract complex features and classify data effectively, outperforming traditional statistical and shallow Machine Learning (ML) approaches [35, 37, 38]. Among the most prominent methods are Deep Neural Networks (DNN) for network traffic classification [38], Convolutional Neural Networks (CNN) for the analysis of data patterns, and Recurrent Neural Networks (RNN) and their variants Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU), which specialize in the analysis of data

sequences. In addition, techniques such as semi-supervised learning and transfer learning improve the accuracy and robustness of models in the face of sparse labeled data and evolving attack patterns [39, 40].

Deep learning models have demonstrated high accuracy in detecting DDoS attacks. For example, a hybrid model combining CNN and LSTM achieved 99.20 % accuracy on the NSL-KDD dataset [41]. Another study reported an accuracy of 99.63 % using a Deep Feed Forward (DFF) model [42]. Techniques such as tuned vector convolutional deep convolutional neural network (TVCDNN) optimize the structure and parameters of deep neural networks, further improving detection performance [43].

The effectiveness of deep learning models depends on the quality and completeness of the datasets used for training and evaluation [44]. On the other hand, ensuring that deep learning models can operate in real time and adapt to evolving attack patterns remains a critical challenge [45].

In summary, this study advances DDoS detection research by operationalizing an end-to-end pipeline that links-controlled traffic generation, transparent labeling criteria (baseline vs. stress condition), and model training into actionable detection outputs that can inform mitigation decisions (e.g., activation of rate-limiting and scaling policies). In methodological terms, we introduce an application-oriented hybrid CNN–LSTM approach designed to capture both local patterns and temporal dependencies from traffic logs and complementary service-side indicators, under an isolated laboratory testbed that avoids exposure of production infrastructure. Finally, we report an empirical evaluation using a labeled dataset of 6,000 instances under a consistent train/validation/test protocol, providing a reproducible basis for comparing learning-based detection performance in realistic operational conditions.

2. METHOD

2.1. Study design and experimental workflow

A four-stage process was followed: (1) collection and labeling of traffic in a controlled environment supplemented with open-source logs, (2) preprocessing (normalization, deduplication, balancing) and feature engineering, (3) hybrid CNN–LSTM training with model selection on a validation split and final evaluation on a held-out test split, and (4) a deployment-oriented design in which the detector can be integrated with mitigation actions (e.g., filtering/rate limiting and scaling policies) based on detection outcomes.

An experimental methodology was adopted to design, implement, and evaluate the proposed DDoS detection approach under controlled conditions. The study combines (i) a structured review of recent advances in DDoS detection and (ii) an empirical evaluation using a sandboxed test environment, with emphasis on methodological reproducibility, clear labeling, and quantitative performance assessment.

A laboratory testbed was configured to generate traffic scenarios representing both baseline (benign) access patterns and stress conditions intended to emulate DDoS-like behavior at the application layer. Traffic intensity was varied systematically across runs using predefined workload levels, while network traces and service-side indicators were captured to support preprocessing, labeling, and model evaluation. All experiments were conducted exclusively within an isolated environment; endpoints and identifiers were anonymized, and no production infrastructure was involved.

In accordance with responsible security disclosure, implementation-specific operational details (e.g., tool commands, reserved domains, tunneling configuration parameters, or step-by-step deployment instructions) are intentionally omitted. The manuscript therefore focuses on experimental design, labeling strategy, and detection performance rather than on operational procedures.

For training and evaluation, we used a dataset of 6,000 labeled network-traffic logs collected from open sources and controlled experiments in a sandboxed environment. Preprocessing included normalization, removal of duplicates, and class balancing via over/under-sampling. Resampling was applied only to the training split to prevent leakage. After preprocessing, the final dataset contained $N = 6,000$ samples, split into training ($n_{\text{train}} = 4,800$; 80%), validation ($n_{\text{val}} = 600$; 10%), and testing ($n_{\text{test}} = 600$; 10%). The validation split was used for model selection and hyperparameter tuning, and the test split was reserved exclusively for final performance reporting. Feature engineering was then applied to derive the most informative variables for discriminating baseline versus stress-condition traffic.

2.2. Dataset, sources and labeling

Dataset source and provenance. The open-source portion of the dataset was derived from the CICDDoS2019 repository released by the Canadian Institute for Cybersecurity (UNB), which provides labeled benign and DDoS traffic along with flow-level features and PCAPs [46]. Controlled traces were additionally generated in our sandboxed testbed (Section 3.2) to reflect baseline and stress-condition behavior under our experimental setup. From these sources, we constructed a curated dataset of $N = 6,000$ labeled instances under a consistent train/validation/test split, following the preprocessing steps described above.

Access information. The CICDDoS2019 dataset is publicly available from UNB (Canadian Institute for Cybersecurity) [46]. To facilitate verification and replication, the curated subset definition (sampling

criteria), feature schema, and preprocessing scripts can be provided as supplementary material or upon reasonable request, consistent with responsible disclosure constraints.

2.3. Hybrid CNN–LSTM model and mathematical formulation

Problem formulation. Let $X \in \mathbb{R}^{T \times d}$ denote a traffic sample represented as a sequence of T time steps with d engineered features per step (derived from network traces and service-side indicators). The detection task is formulated as binary classification $y \in \{0,1\}$, where $y = 1$ denotes Attack (stress condition) and $y = 0$ denotes Benign (baseline condition).

Convolutional feature extraction (CNN block). A 1-D convolution is applied along the temporal axis to capture local patterns in the traffic sequence. For filter k with kernel W_k and bias b_k , the feature map is computed as

$$z_k = \phi(W_k * X + b_k), \quad (1)$$

where $*$ denotes 1-D convolution and $\phi(\cdot)$ is a nonlinear activation function. A pooling operator $g(\cdot)$ is then used to reduce dimensionality and improve invariance, producing $\tilde{z}_k = g(z_k)$. Stacking K filters yields $\tilde{Z} \in \mathbb{R}^{T' \times K}$.

Temporal dependency modeling (LSTM block). The pooled sequence $\tilde{Z}$ is passed to an LSTM to capture longer-range dependencies. At time t , with input $\tilde{z}_t$, the LSTM updates are

$$i_t = \sigma(W_i \tilde{z}_t + U_i h_{t-1} + b_i), f_t = \sigma(W_f \tilde{z}_t + U_f h_{t-1} + b_f), \quad (2)$$

$$\tilde{c}_t = \tanh(W_c \tilde{z}_t + U_c h_{t-1} + b_c), c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t, \quad (3)$$

$$o_t = \sigma(W_o \tilde{z}_t + U_o h_{t-1} + b_o), h_t = o_t \odot \tanh(c_t), \quad (4)$$

where $\sigma(\cdot)$ is the logistic sigmoid and $\odot$ is the Hadamard product.

Classification head and decision rule. The final hidden state (or an aggregation of $\{h_t\}$) is mapped to a probability of attack via

$$p = \sigma(w^\top h + b), \quad (5)$$

and the predicted label is $\hat{y} = 1$ if $p \geq \tau$ (with $\tau = 0.5$ unless otherwise stated), otherwise $\hat{y} = 0$.

Training objective. The model is trained by minimizing the binary cross-entropy loss with model selection performed on the validation split and final reporting on the held-out test split.

$$\mathcal{L} = -\frac{1}{N} \sum_{n=1}^N [y_n \log(p_n) + (1 - y_n) \log(1 - p_n)], \quad (6)$$

Implementation note. Exact layer sizes (e.g., number of filters/units), kernel lengths, and regularization settings are reported in the experiment configuration files and can be provided as supplementary material to support full reproducibility.

Training protocol and reproducibility. The hybrid CNN–LSTM was trained under a supervised binary-classification setting using the training split, with model selection performed on the validation split and final reporting on the held-out test split. To mitigate overfitting, training was monitored using validation performance criteria (e.g., validation loss/metrics) and stopped once convergence was observed. Hyperparameters such as the optimizer choice, learning-rate schedule, batch size, and maximum epochs were set prior to final testing based on validation performance. Because the present manuscript prioritizes experimental design and defensible reporting, the complete training configuration (including exact hyperparameter values and implementation settings) is available in the experiment scripts/logs and can be provided as supplementary material or upon reasonable request.

3. RESULTS AND DISCUSSION

3.1. Experimental Setup Overview

The experimental setup consisted of three functional components: (i) a traffic-generation client that produced baseline and stress workloads, (ii) a sandboxed target web service that handled incoming requests and served as the monitored system under test, and (iii) a monitoring and capture component responsible for collecting network traces and service-side indicators during each run. Workload levels were predefined to enable repeatable comparisons across scenarios. All endpoints were anonymized, and the environment was isolated from production systems.

Figure 1 summarizes the experimental testbed and the roles of each component. Under stress conditions, the traffic-generation client produced a controlled high-rate application-layer workload directed at

the sandboxed web service, while the monitoring component captured network traces and service-side indicators for labeling and evaluation. To prevent operational misuse, implementation-specific procedures are not described; instead, the manuscript focuses on experimental design, labeling strategy, and quantitative detection performance.

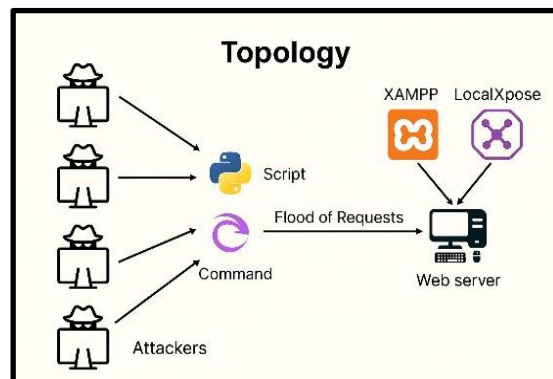


Figure 1. DDoS Attack Topology

Figure 2 shows the local service-administration interface used to deploy and monitor the sandboxed web service during baseline and stress runs. This interface supported routine availability checks and facilitated trace collection; however, tool-specific configuration details are omitted to maintain focus on experimental design, labeling, and detection performance.

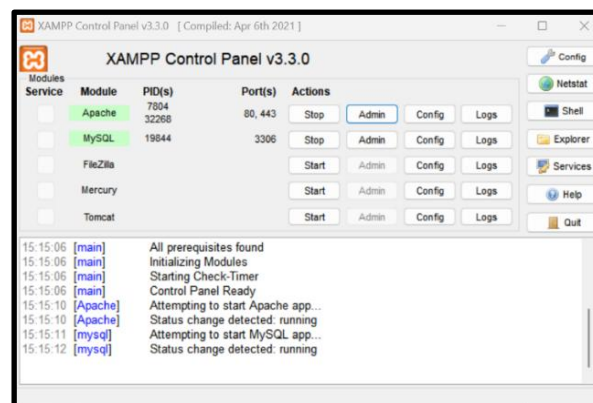


Figure 2. Xampp control panel v3.3.0

The test bench was designed to generate HTTP traffic in a controlled manner to an experimental web service, systematically varying the load intensity using predefined concurrency levels. This strategy allowed us to build repeatable traces for preprocessing, tagging, and performance evaluation of the proposed detection approach. Consequently, the manuscript prioritizes the description of the experimental design, labeling criteria, and performance metrics, rather than tool-level implementation procedures or commands.

To enable controlled access to the experimental service during traffic capture, a secure connectivity mechanism was employed within an isolated testing environment (sandbox). All endpoints and identifiers were anonymized, no production infrastructure was used, and configuration details were deliberately omitted, in accordance with responsible disclosure principles and to prevent operational misuse. The experimental service was administered through the local server management interface to verify service availability and support trace collection throughout the runs.

As illustrated in Figure 3, the service experienced a marked degradation under the stress condition, including elevated error responses and periods of unavailability consistent with resource saturation. This behavior supports the experimental labeling of stress runs and highlights the operational impact of high-rate workloads on service continuity in the sandboxed environment.

Additionally, monitoring indicators collected during the same runs (e.g., increased timeout events and abnormal response-status patterns) corroborated the observed degradation and were used to contextualize the captured traces for subsequent preprocessing and evaluation.

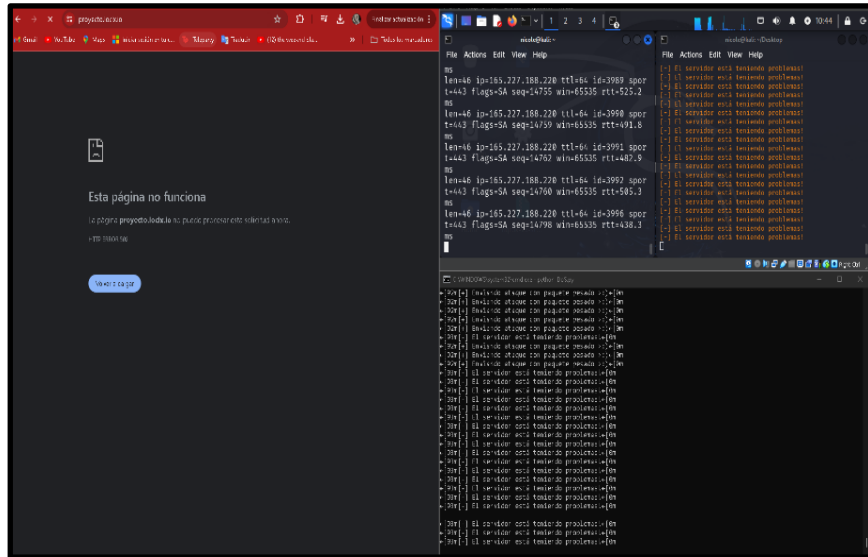


Figure 3. Server downtime by DDoS attack

3.2. Traffic Generation and Data Capture Protocol

The 80/10/10 train/validation/test split ($n_{\text{train}} = 4,800$; $n_{\text{val}} = 600$; $n_{\text{test}} = 600$) and resampling policy are described in the Method section.

To support training and evaluation of the proposed detection approach, network traffic was generated and captured under controlled conditions in an isolated laboratory testbed. The experimental service was deployed in a sandboxed environment and subjected to multiple traffic conditions, including baseline (benign) access patterns and stress scenarios intended to emulate DDoS-like behavior at the application layer. Traffic intensity was varied systematically across runs using predefined workload levels to ensure repeatability and enable comparative analysis. No production infrastructure was used at any stage of the experiments.

During each run, network traces and service-side indicators were recorded to enable downstream preprocessing, labeling, and feature construction. Label assignment followed the experimental schedule (baseline vs. stress condition) and was cross-checked against observed service behavior (e.g., response-status patterns and degradation signals) to improve consistency of ground-truth annotation. The resulting dataset was then prepared for modeling through standard preprocessing steps (data cleaning, normalization where applicable, and transformation into model-ready representations).

3.3. Analysis and Interpretation

Impact of stress condition. The stress runs produced a substantial increase in request pressure compared with baseline traffic, leading to measurable service degradation and intermittent unavailability. These outcomes are consistent with prior findings that high-rate application-layer workloads can exhaust server-side resources and reduce service responsiveness. In our setup, workload intensity was controlled through predefined levels to enable consistent comparisons and to support systematic evaluation of the proposed detection method across multiple operating points.

Service administration and monitoring. The experimental web service was administered through a local server management interface to verify service availability and to support trace collection throughout baseline and stress conditions. During each run, server-side indicators (e.g., response status patterns and error rates) were monitored to contextualize service degradation and to complement the network traces used for labeling and evaluation.

Independent monitoring indicators collected from the traffic-generation and capture components corroborated the stress condition, showing a sustained increase in transmission activity and a corresponding rise in service-side degradation signals (e.g., timeouts and abnormal response-status patterns). These observations were used only to contextualize the captured traces and to support consistent labeling of stress runs; the analysis prioritizes measurable performance outcomes of the detection models rather than operational implementation details.

From an operational perspective, the proposed detection capability can support broader risk-management frameworks in critical sectors (e.g., financial services) by enabling earlier activation of mitigation controls and improving service continuity [47].

3.4. Comparisons with other algorithms

Prior studies have reported competitive performance for several learning algorithms in DDoS detection, including Random Forest, KNN, SVM variants, and deep learning models such as LSTM and feed-forward networks [48–51]. However, direct comparison across studies is often constrained by differences in datasets, labeling procedures, feature representations, and experimental protocols.

To strengthen the comparative discussion, Table 1 summarizes the datasets and performance metrics reported by selected prior studies referenced in this section. Because these works differ in traffic representations, labeling strategies, and evaluation protocols, the table is intended as contextual benchmarking rather than a strict head-to-head comparison.

Table 1. Comparison with other algorithms

Ref.	Model / Approach	Dataset(s) reported	Task	Evaluation protocol (as reported)	Reported performance (selected)
[48]	RF vs KNN vs Logistic Regression (KNN modified)	NSL-KDD	Binary classification (Normal vs DDoS)	k-fold / cross-validation	RF $\approx$ 99% acc; KNN $\approx$ 96% acc; LR $\approx$ 85% acc
[49]	Random Forest (compared vs J48, NB, SVM, KNN, MLP)	NSL-KDD (test file: 22,544 instances)	IDS traffic classification (legitimate vs anomalous/DDoS)	WEKA; 10-fold cross-validation; default settings	Acc 98.70%; FP rate 1.4%; Precision 98.7%; Recall 98.7%
[50]	Hybrid N-Gram heuristic feature selection + SVM/KNN/NN	CIC-2017; MIB-Dataset 2016; H2N-Payload Dataset	DDoS detection (payload/HTTP focus)	Payload-based; protocol focus differs from flow/traffic-log representations	Best result reported: SVM with 4-Gram accuracy up to 99.86–100% (dataset-dependent)
[51]	LSTM and FFN for EV charging station DDoS	CICEV2023 dataset	DDoS detection in EV charging infrastructure	Domain differs (EV station telemetry/metrics)	Study reports LSTM advantages in speed; FFN higher accuracy
This work	Hybrid CNN–LSTM	CICDDoS2019 (open-source) + controlled sandbox traces	Binary (Attack vs Benign)	Unified 80/10/10 split; confusion matrix provided	Acc 98.67%; Prec 98.01%; Rec 96.73%; F1 97.37%

In our experiments, the proposed hybrid CNN–LSTM model was evaluated using standard detection metrics (accuracy, precision, recall, and F1-score). The model achieved 98.67% accuracy, 98.01% precision, 96.73% recall, and an F1-score of 97.37%, indicating reliable discrimination between baseline and stress-condition traffic in the evaluated dataset. Under the same experimental setting, Random Forest achieved 96.42% accuracy, while a standalone LSTM achieved 97.10% accuracy. Overall, these results suggest that combining convolutional and recurrent components improves performance over the tested baselines within our controlled evaluation. Here, ‘Attack’ is treated as the positive class and ‘Benign’ as the negative class. To provide class-wise error transparency, Table 2 reports the confusion matrix of the proposed hybrid model on the held-out test set.

Table 2. Confusion matrix of the proposed hybrid CNN–LSTM on the test set ($n_{\text{test}} = 600$)

	Pred: Attack	Pred: Benign
Actual: Attack	TP = 148	FN = 5
Actual: Benign	FP = 3	TN = 444

To summarize the comparative performance under the same evaluation protocol, Table 3 reports the test-set metrics for the proposed hybrid CNN–LSTM model and the baseline methods.

Table 3. Performance comparison with baseline models (test set).

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
Hybrid CNN–LSTM (proposed)	98.67	98.01	96.73	97.37
Random Forest	96.42	—	—	—
LSTM (standalone)	97.10	—	—	—

4. CONCLUSION

This study presents a controlled and reproducible testbed together with a hybrid CNN–LSTM model for DDoS detection, designed to capture both spatial feature interactions and temporal dependencies in network traffic. To address practical constraints such as limited labeled samples and traffic drift, the framework is positioned to incorporate semi-supervised learning and transfer learning during model refinement. Empirical

evaluation on 6,000 labeled logs shows that the proposed model attains 98.67% accuracy, 98.01% precision, 96.73% recall, and 97.37% F1-score, outperforming Random Forest and a standalone LSTM under the same evaluation protocol. Overall, the results support the feasibility of deploying hybrid deep learning detectors as part of a broader mitigation pipeline for protecting critical services against DDoS disruptions.

As a recommendation, DDoS mitigation can be strengthened through access control lists (ACL) to regulate inbound traffic, as well as by improving network and server capacity. In practice, this entails ensuring sufficient bandwidth and redundancy at the hosting level and enabling rapid scalability through load balancing and elastic compute resources to maintain service availability during attack peaks.

For future research, we propose validating the approach in production environments with real traffic and incorporating mechanisms that improve operational applicability, such as explainable AI (XAI) for transparent decision-making and privacy-preserving training schemes (e.g., federated learning) for distributed deployments.

REFERENCES

- [1] R. Maciel, J. Araujo, C. Melo, P. Pereira, J. Dantas, and P. Maciel, *Impact evaluation of DDoS and Malware attack using IoT devices*. 2021. doi: 10.1515/9783110619751-001.
- [2] Abhishta, R. Joosten, and L. J. M. Nieuwenhuis, "Comparing alternatives to measure the impact of DDoS attack announcements on target stock prices," *J Wirel Mob Netw Ubiquitous Comput Dependable Appl*, vol. 8, no. 4, pp. 1–18, 2017, doi: 10.22667/JOWUA.2017.12.31.001.
- [3] M. Sachdeva, G. Singh, and K. Kumar, "An emulation based impact analysis of DDoS attacks on web services during flash events," in *2011 2nd International Conference on Computer and Communication Technology, ICCCT-2011*, 2011, pp. 479–484. doi: 10.1109/ICCCT.2011.6075134.
- [4] N. Kathirkamanathan, B. Thevarasa, G. Mahadevan, N. Skandhakumar, and N. Kuruwitaarachchi, "Prevention of DDoS Attacks Targeting Financial Services using Supervised Machine Learning and Stacked LSTM," in *2022 IEEE 7th International conference for Convergence in Technology, I2CT 2022*, 2022. doi: 10.1109/I2CT54291.2022.9825228.
- [5] R. P. Janivasya and I. D. A. Rachmawati, "DDoS Detection using Machine Learning Approach," in *Procedia Computer Science*, 2024, pp. 1157–1164. doi: 10.1016/j.procs.2024.10.345.
- [6] H. Guesmi, A. Kalghoum, R. Guesmi, and L. A. Saïdane, *SMC: A New Strategy Based on Software-Defined Networking to Mitigate the Impact of Anomalies on Cooperative Cloud*, vol. 12983 LNCS. 2021. doi: 10.1007/978-3-030-88207-5_14.
- [7] F. A. Olivos Estrada, W. S. Flores-Sotelo, C. R. Barrera-Avalos, W. A. Martínez-Aberga, R. Sulca-Guillen, and J. M. Chávez-Díaz, "Evaluating the impact of risk management and cybersecurity on decision-making in the Peruvian National Informatics System," *Bulletin of Electrical Engineering and Informatics*, vol. 14, no. 5, pp. 3816–3830, Oct. 2025, doi: 10.11591/eei.v14i5.9781.
- [8] J. O. Nehinbe and U. S. Onyebor, "An exhaustive study of DDOS attacks and DDOS datasets," *International Journal of Internet Technology and Secured Transactions*, vol. 10, no. 3, pp. 268–285, 2020, doi: 10.1504/IJTST.2020.107075.
- [9] D. T. Bennet, P. S. Bennet, and D. Anitha, "Securing Smart City Networks - Intelligent Detection of DDoS Cyber Attacks," in *Proceedings of 5th International Conference on Contemporary Computing and Informatics, IC3I 2022*, 2022, pp. 1575–1580. doi: 10.1109/IC3I56241.2022.10073271.
- [10] S. Huang, "Proposing a self-help privilege for victims of cyber attacks," *George Washington Law Review*, vol. 82, no. 4, pp. 1229–1266, 2014.
- [11] F. S. Alabbadi, E. M. Al Amaren, and S. I. Aletein, "International Responsibility Arising from Cyberattacks in The Light of the Contemporary International Law," *International Journal of Cyber Criminology*, vol. 16, no. 1, pp. 156–169, 2022, doi: 10.5281/zenodo.4766562.
- [12] A. N. H. Dhatreesh Sai, B. H. Tilak, N. Sai Sanjith, P. Suhas, and R. Sanjeetha, "Detection and Mitigation of Low and Slow DDoS attack in an SDN environment," in *2022 IEEE International Conference on Distributed Computing, VLSI, Electrical Circuits and Robotics, DISCOVER 2022 - Proceedings*, 2022, pp. 106–111. doi: 10.1109/DISCOVER55800.2022.9974724.
- [13] S. Kumar, A. Guerrero, and C. Navarro, "Cyber Security Flood Attacks and Risk Assessment for Internet of Things (IoT) Distributed Systems," in *2023 IEEE World AI IoT Congress, AIIoT 2023*, 2023, pp. 392–397. doi: 10.1109/AIIoT58121.2023.10174553.
- [14] L. Huraj, M. Šimon, and J. Lietava, "Dataset of DDoS attacks on Fibaro home center 3 for smart home security," *Data Brief*, vol. 57, 2024, doi: 10.1016/j.dib.2024.110991.
- [15] H. Khandare, S. Jain, and R. Doriya, *A Survey on HTTP Flooding—A Distributed Denial of Service Attack*, vol. 475. 2023. doi: 10.1007/978-981-19-2840-6_4.
- [16] L. Kavisankar and C. Chellappan, *T-RAP: (TCP Reply Acknowledgement Packet) a resilient filtering model for DDoS attack with spoofed IP address*, vol. 197 CCIS. 2011. doi: 10.1007/978-3-642-22543-7_14.
- [17] B. Ramesh and S. J. J. Thangaraj, "Distributed Denial of Service (DDoS) in Internet of Things (IoT) Environment: Classification and Overview," in *AIP Conference Proceedings*, 2024. doi: 10.1063/5.0219386.
- [18] N. Hoque, D. K. Bhattacharyya, and J. K. Kalita, "Botnet in DDoS Attacks: Trends and Challenges," *IEEE Communications Surveys and Tutorials*, vol. 17, no. 4, pp. 2242–2270, 2015, doi: 10.1109/COMST.2015.2457491.

- [19] K. Soni and S. Singh, "A proposed DoS detection scheme for mitigating DoS attack," *International Journal of Emerging Technology and Advanced Engineering*, vol. 10, no. 4, pp. 172–179, 2020.
- [20] L. Gondi, S. Sambangi, P. K. Priya, and S. S. Anjum, "A Machine Learning Approach for DDoS Attack Detection in CIC-DDoS2019 Dataset Using Multiple Linear Regression Algorithm," in *Springer Proceedings in Mathematics and Statistics*, 2024, pp. 393–403. doi: 10.1007/978-3-031-51167-7_38.
- [21] M. A. Ribeiro, M. S. Pereira Fonseca, and J. de Santi, "Detecting and mitigating DDoS attacks with moving target defense approach based on automated flow classification in SDN networks," *Comput Secur*, vol. 134, 2023, doi: 10.1016/j.cose.2023.103462.
- [22] Q. Shafi and A. Basit, "DDoS Botnet Prevention using Blockchain in Software Defined Internet of Things," in *Proceedings of 2019 16th International Bhurban Conference on Applied Sciences and Technology, IBCAST 2019*, 2019, pp. 624–628. doi: 10.1109/IBCAST.2019.8667147.
- [23] M. Najafimehr, S. Zarifzadeh, and S. Mostafavi, "A hybrid machine learning approach for detecting unprecedented DDoS attacks," *J Supercomput*, vol. 78, no. 6, pp. 8106–8136, Apr. 2022, doi: 10.1007/s11227-021-04253-x.
- [24] R. K. Batchu and H. Seetha, "A generalized machine learning model for DDoS attacks detection using hybrid feature selection and hyperparameter tuning," *Computer Networks*, vol. 200, p. 108498, Dec. 2021, doi: 10.1016/j.comnet.2021.108498.
- [25] S. Sumathi, R. Rajesh, and N. Karthikeyan, "DDoS Attack Detection Using Hybrid Machine Learning Based IDS Models," *J Sci Ind Res (India)*, vol. 81, no. 3, pp. 276–286, 2022, Accessed: Dec. 12, 2025. [Online]. Available: <https://nopr.niscares.in/handle/123456789/59310>
- [26] Y. Wei, J. Jang-Jaccard, F. Sabrina, A. Singh, W. Xu, and S. Camtepe, "AE-MLP: A Hybrid Deep Learning Approach for DDoS Detection and Classification," *IEEE Access*, vol. 9, pp. 146810–146821, 2021, doi: 10.1109/ACCESS.2021.3123791.
- [27] D. Alghazzawi, O. Bamasag, H. Ullah, and M. Z. Asghar, "Efficient Detection of DDoS Attacks Using a Hybrid Deep Learning Model with Improved Feature Selection," *Applied Sciences*, vol. 11, no. 24, p. 11634, Dec. 2021, doi: 10.3390/app112411634.
- [28] A. A. Elsaedy, A. Jamalipour, and K. S. Munasinghe, "A Hybrid Deep Learning Approach for Replay and DDoS Attack Detection in a Smart City," *IEEE Access*, vol. 9, pp. 154864–154875, 2021, doi: 10.1109/ACCESS.2021.3128701.
- [29] N. Kadam and K. R. Sekhar, "Machine Learning Approach of Hybrid KSVN Algorithm to Detect DDoS Attack in VANET," *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 7, pp. 718–722, 2021, doi: 10.14569/IJACSA.2021.0120782.
- [30] H. Elubeyd and D. Yiltas-Kaplan, "Hybrid Deep Learning Approach for Automatic DoS/DDoS Attacks Detection in Software-Defined Networks," *Applied Sciences*, vol. 13, no. 6, p. 3828, Mar. 2023, doi: 10.3390/app13063828.
- [31] A. F. Al-zubidi, A. K. Farhan, and S. M. Towfek, "Predicting DoS and DDoS attacks in network security scenarios using a hybrid deep learning model," *Journal of Intelligent Systems*, vol. 33, no. 1, Apr. 2024, doi: 10.1515/jisys-2023-0195.
- [32] N. U. Ain, M. Sardaraz, M. Tahir, M. W. Abo Elsoud, and A. Alourani, "Securing IoT Networks Against DDoS Attacks: A Hybrid Deep Learning Approach," *Sensors*, vol. 25, no. 5, p. 1346, Feb. 2025, doi: 10.3390/s25051346.
- [33] S. Wang, C. Ma, H. Gao, D. Deng, C. Fernandez, and F. Blaabjerg, "Improved hyperparameter Bayesian optimization-bidirectional long short-term memory optimization for high-precision battery state of charge estimation," *Energy*, vol. 328, p. 136598, Aug. 2025, doi: 10.1016/j.energy.2025.136598.
- [34] A. Saied, R. E. Overill, and T. Radzik, "Detection of known and unknown DDoS attacks using Artificial Neural Networks," *Neurocomputing*, vol. 172, pp. 385–393, Jan. 2016, doi: 10.1016/j.neucom.2015.04.101.
- [35] M. Mittal, K. Kumar, and S. Behal, "Deep learning approaches for detecting DDoS attacks: a systematic review," *Soft comput*, vol. 27, no. 18, pp. 13039–13075, Sep. 2023, doi: 10.1007/s00500-021-06608-1.
- [36] M. Subramanian, K. S. Vadivel, and R. Sowmya, "Performance Evaluation of Deep Learning Models in Detection of Distributed Denial of Service Attacks," in *Proceedings of the 5th International Conference on Electronics, Communication and Aerospace Technology, ICECA 2021*, 2021, pp. 652–658. doi: 10.1109/ICECA52323.2021.9676091.
- [37] M. Subrmanian, K. Shanmugavadivel, P. S. Nandhini, and R. Sowmya, *Evaluating the Performance of LSTM and GRU in Detection of Distributed Denial of Service Attacks Using CICDDoS2019 Dataset*, vol. 140. 2022. doi: 10.1007/978-981-19-2948-9_38.
- [38] S. Malliga, S. V. Kogilavani, and R. Sowmya, "Deep discover: Deep learning models for detecting distributed denial of service (DDOS) attacks," in *AIP Conference Proceedings*, 2022. doi: 10.1063/5.0074445.
- [39] A. S. A. Issa and Z. Albayrak, "DDoS Attack Intrusion Detection System Based on Hybridization of CNN and LSTM," *Acta Polytechnica Hungarica*, vol. 20, no. 2, pp. 105–123, 2023, doi: 10.12700/APH.20.2.2023.2.6.
- [40] H. Mennour and S. Mostefai, "Deep learning-based distributed denial-of-service detection," *International Journal of Networking and Virtual Organisations*, vol. 26, no. 1–2, pp. 80–103, 2022, doi: 10.1504/IJNVO.2022.121921.
- [41] A. S. A. Issa and Z. Albayrak, "DDoS Attack Intrusion Detection System Based on Hybridization of CNN and LSTM," *Acta Polytechnica Hungarica*, vol. 20, no. 2, pp. 105–123, 2023, doi: 10.12700/APH.20.2.2023.2.6.
- [42] P. Khuphiran, P. Leelaprute, P. Uthayopas, K. Ichikawa, and W. Watanakesuntorn, "Performance comparison of machine learning models for DDoS attacks detection," in *2018 22nd International Computer Science and Engineering Conference, ICSEC 2018*, 2018. doi: 10.1109/ICSEC.2018.8712757.

- [43] N. G. B. Amma and S. Selvakumar, "Optimization of vector convolutional deep neural network using binary real cumulative incarnation for detection of distributed denial of service attacks," *Neural Comput Appl*, vol. 34, no. 4, pp. 2869–2882, 2022, doi: 10.1007/s00521-021-06565-8.
- [44] B.-S. Tran, T.-H. Ho, T.-X. Do, and K.-H. Le, *Empirical Performance Evaluation of Machine Learning based DDoS Attack Detections*, vol. 215. 2022. doi: 10.1007/978-3-030-90119-6_23.
- [45] C. Li *et al.*, "Detection and defense of DDoS attack–based on deep learning in OpenFlow-based SDN," *International Journal of Communication Systems*, vol. 31, no. 5, 2018, doi: 10.1002/dac.3497.
- [46] Canadian Institute for Cybersecurity (CIC), "DDoS evaluation dataset (CIC-DDoS2019)," 2019. [Online]. Available: <https://www.unb.ca/cic/datasets/ddos-2019.html>
- [47] K. Chlouverakis, "How artificial intelligence is reshaping the financial services industry." Accessed: Jan. 02, 2025. [Online]. Available: https://www.ey.com/en_gr/insights/financial-services/how-artificial-intelligence-is-reshaping-the-financial-services-industry.
- [48] S. Haribalaji and P. Ranjana, "Distributed Denial of Service (DDoS) Attack Detection Using Classification Algorithm," in *2024 International Conference on Advances in Data Engineering and Intelligent Computing Systems (ADICS)*, IEEE, Apr. 2024, pp. 1–6. doi: 10.1109/ADICS58448.2024.10533510.
- [49] Z. F. Faruq, T. Mantoro, M. A. Catur Bhakti, and Wandy, "Random Forest Classifier Evaluation in DDoS Detection System for Cyber Defence Preparation," in *2022 IEEE 8th International Conference on Computing, Engineering and Design (ICCED)*, IEEE, Jul. 2022, pp. 1–5. doi: 10.1109/ICCED56140.2022.10010341.
- [50] A. Maslan, K. M. Bin Mohamad, A. Hamid, H. Pangaribuan, and S. Sitohang, "Feature Selection to Enhance DDoS Detection Using Hybrid N-Gram Heuristic Techniques," *JOIV: International Journal on Informatics Visualization*, vol. 7, no. 3, pp. 815–822, Sep. 2023, doi: 10.30630/joiv.7.3.1533.
- [51] Y. A. Anli, Z. Ciplak, M. Sakaliuzun, S. Z. Izgu, and K. Yildiz, "DDoS detection in electric vehicle charging stations: A deep learning perspective via CICEV2023 dataset," *Internet of Things*, vol. 28, p. 101343, Dec. 2024, doi: 10.1016/j.iot.2024.101343.

BIOGRAPHIES OF AUTHORS



Josías Axel Chávez-Campoverde  Pre-Bachelor in Systems Engineering from Universidad de Lima. Experienced in programming language and commercial software. Analyst in scientific research. He can be contacted at email: 20210614@aloe.ulima.edu.pe



Misael Adrián Chávez-Campoverde  Bachelor in Multimedia Audiovisual Communication (CAM) at Toulouse Lautrec. Experienced in digital marketing, photography, motion graphics in the financial, real estate and creative fields. Analyst in scientific research. He can be contacted at email: misaelch7vez@gmail.com



Daniel André Chávez-Campoverde . Bachelor's Degree in Business Administration from Universidad Tecnológica del Peru. Specialization in Business Analytics at Universidad del Pacífico. Graduate Program at Centennial College, Toronto. He can be contacted at email: dchavezc@my.centennialcollege.ca



Naomi Alessandra Chavez-Campoverde . Bachelor's Degree in Business Administration from the University of Lima, with a specialization in Innovation and Technology. She furthered her academic journey by completing Post-Degree Diplomas in Data Analytics and Sales at Douglas College, Canada, where she gained in-depth knowledge of data-driven decision-making and market dynamics within the Canadian business environment. She can be contacted at email: chavezcampovern@student.douglascollege.ca



Prof. Jorge Miguel Chávez-Díaz    PhD in Accounting from Universidad Nacional Federico Villarreal. Master in Accounting (mention in Financial Auditing) by ULADECH. Graduate professor en Universidad Peruana de Ciencias Aplicadas. Associate Professor at the undergraduate and postgraduate at the Universidad Nacional Mayor de San Marcos. Consultant in accounting and financial issues, with the use of CAATTs. Lecturer in Accounting - Systems and Information Technology. Winner of the National Accounting Award 2023 JDCCPP in Perú. Expert in design and conceptual review for research. He has more than twelve publications in journals indexed in Scopus. He can be contacted at email: pejochav@upc.edu.pe