Journal of Artificial Intelligence and Technology, 2026, 6, 207-214
https://doi.org/10.37965/jait.2025.0887

ISTP
RESEARCH ARTICLE

CLSTMNet Architecture: A CNN–LSTM-Based Hybrid Deep Learning Model for DDoS Attack Detection and Mitigation in Network Security

Danang[1] **and Zaenal Mustofa**[2]
[1]Universitas Sains dan Teknologi Komputer (STEKOM), Semarang, Jalan Majapahit No 605, Semarang, 50192, Jawa Tengah, Indonesia
[2]Universitas Negeri Yogyakarta, Yogyakarta, Indonesia

(Received 31 July 2025; Revised 14 November 2025; Accepted 19 December 2025; Published online 15 January 2026)

Abstract: Distributed denial-of-service (DDoS) attacks represent one of the most damaging cybersecurity threats to modern network systems. The impact of this attack causes server failure and creates complaints about service inconvenience from users, thus reducing the company's reputation and trust; more crucially, it is the loss of revenue. Although intrusion detection systems (IDSs) and other conventional security mechanisms have been widely deployed, many advanced DDoS attacks continue to bypass these defenses due to their evolving and complex patterns. This study aims to provide a state-of-the-art strategy to identify denial-of-service (DDoS) attacks more precisely using machine learning (ML) calculations. Creation of a modern deep learning (DL) strategy identifies DDoS attacks more precisely by combining the two best DL calculations and comparing their execution by actualizing them on the most challenging dataset. This research applies a combination strategy of two DL calculations models, convolutional neural network (CNN) and long short-term memory (LSTM). These calculations are actualized on the Network Security Laboratory–Knowledge Discovery and Data Mining (NSL-KDD) dataset, which is considered the most challenging dataset for DDoS attack discovery. The results show that the modern DL strategy created in this consideration outperforms other state-of-the-art strategies in terms of precision and discovery rate. The combination of CNN and LSTM results in superior execution than either calculation alone. This implies that the modern DL strategy created in this consideration is a feasible approach to identify DDoS attacks with high precision.

Keywords: CNN; DDoS; deep learning; machine learning

I. INTRODUCTION

Global internet usage has increased significantly in recent decades, with the number of users reaching around 5.5 billion in December 2024 [1]. The massive development of digital technology has created risks and threats to computer security that are also becoming increasingly relevant, including denial of service (DoS) and distributed denial-of-service (DDoS). Attackers are able to exploit vulnerabilities quickly after they enter the network. One of the tools used to protect computer systems and networks is the intrusion detection system (IDS), which is able to analyze and identify any type of suspicious or unwanted activity that may occur within a computer network. Reference [2] stated that misclassification due to low attack detection accuracy and the inability to identify modern attacks are the main focus of IDS.

DoS attacks are well known to be disturbing because they can disrupt online services, from companies to government agencies. With the impact of disrupting business operations and public services, DoS cannot be underestimated. Unlike other types of cyberattacks, DDoS attacks do not attempt to penetrate the security perimeter but make website and server inaccessible to legitimate users. See Fig. 1.

Unlike typical cyber threats, DDoS attacks aim not to breach system defenses but to overload resources—such as bandwidth or memory—making services inaccessible to legitimate users [3]. To overcome the limitations of traditional IDS, researchers have turned to machine learning (ML), which analyzes large-scale network data to detect anomalies [4]. While ML enhances detection capabilities, its performance heavily depends on the size and quality of datasets and often lacks adaptability to complex or evolving threats [5].

Consequently, deep learning (DL) has gained attention for its superior performance in handling intricate data patterns, with successful applications across speech recognition, image processing, and natural language tasks [6]. Despite its advantages, DL models can still struggle with modeling complex relationships among high-dimensional data [7].

This research addresses these challenges by proposing a hybrid DL framework—CLSTMNet—which merges the strengths of convolutional neural networks (CNNs) for feature extraction [8] and long short-term memory (LSTM) for sequential prediction [9]. Designed with seven layers, CLSTMNet aims to deliver high-performance DDoS detection by leveraging the complementary capabilities of CNN and LSTM models.

Complex tasks like image classification and language translation have been effectively addressed through DL, particularly when working with large-scale datasets [10]. In certain scenarios, DL models have even surpassed the performance of human experts. Applying such technology to detect DDoS attacks holds significant promise. However, adapting DL for network intrusion detection introduces unique challenges. One major issue is the

Corresponding author: Danang (e-mail: danang150787@gmail.com).

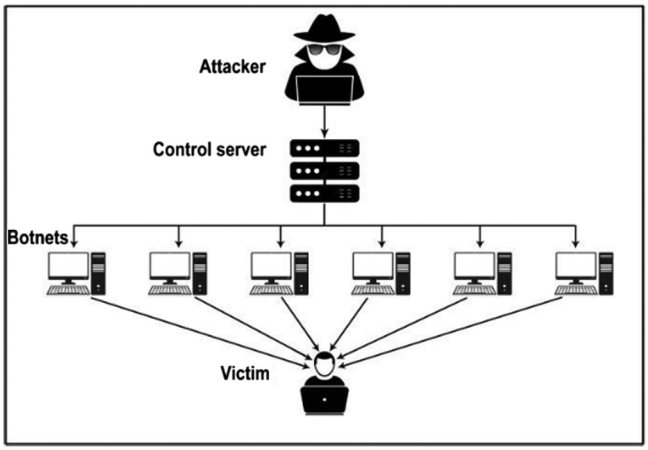

Fig. 1. Structure of DoS and DDoS (Cloudflare (2021).

limited size of training datasets, which results in inadequate learning and ineffective model evaluation. Another critical limitation is that many current DL models either lack high detection accuracy or involve complex computations, thus reducing their overall efficiency. This research aims to directly address both of these challenges.

This study proposes CLSTMNet, a compact and efficient CNN–LSTM hybrid designed to balance high detection accuracy and low computational cost; in contrast to previous hybrid intrusion detection frameworks that rely on multi-branch architectures or ensemble stacking, its key innovation lies in combining spatial and sequential learning into a single efficient model, reducing model complexity while maintaining precision and robustness on the Network Security Laboratory–Knowledge Discovery and Data Mining (NSL-KDD) dataset.

The rest of the paper is structured as follows: Section II provides the related works on DDoS attacks, ML, DL, and IDS; Section III discusses the methodology and the proposed framework model; Section IV discusses the result in detail; and finally, Section V discusses the conclusion.

II. LITERATURE REVIEW

A. RECENT ADVANCES IN ML AND DL TECHNIQUES FOR DDoS DETECTION

Research on IDSs has evolved significantly with the emergence of advanced datasets and ML techniques. One of the most widely used benchmarks in this domain is the NSL-KDD dataset, developed to address the redundancy and imbalance issues found in the earlier KDD Cup 99 dataset. Several studies have demonstrated its effectiveness for evaluating IDS models. For example, feature selection methods such as Subset CFS have been employed to reduce dimensionality while preserving classification performance, with the random forest (RF) algorithm consistently achieving strong accuracy and efficiency in training and testing phases [11].

To enhance detection capabilities, hybrid and ensemble methods have gained increasing attention. A notable approach is the neuro-fuzzy ensemble classifier, which integrates multiple base learners and uses boosting to improve detection rates and computational speed [12]. Likewise, the use of artificial neural networks (ANNs) combined with black hole optimization demonstrated high predictive accuracy in cloud-based DoS detection scenarios [13]. Meanwhile, methods inspired by biological systems, such as the dendritic cell algorithm (DCA) derived from artificial immune systems (AIS), have proven effective in identifying both DoS and DDoS attacks [14].

Feature engineering continues to play a critical role in improving anomaly detection. Studies have shown that examining relationships between network protocols and intrusion types, along with selecting relevant attributes using tools such as WEKA, can lead to better classification results [15,16]. Reference [17] advanced this further by proposing a multi-feature selection strategy involving information gain, co-clustering, and entropy estimation, paired with the Extra-Trees classifier. In a subsequent study, the same authors developed a semi-supervised model incorporating RF, which outperformed traditional approaches by reducing false positives and achieving 93% accuracy [18].

Other notable contributions include the application of genetic algorithms (GAs) for feature selection in conjunction with Bernoulli Naïve Bayes, although this combination yielded modest results [19]. On the other hand, the distributed random forest (DRF) algorithm, when tested using platforms like WEKA and H2O, demonstrated more reliable performance [20]. Reference [21] employed a simple neural network architecture on the Diductor platform, reaching training and testing accuracies above 97%. Similarly, [22] developed a hybrid model integrating mean absolute deviation (MAD) with RF, achieving high accuracy with a minimal false alarm rate.

Further innovations include two-phase detection systems that incorporate client-side preprocessing and proxy-side classification using multiple ML algorithms. These systems have been particularly effective in detecting previously unseen attacks, with RF again yielding the best results [23]. Additionally, [24] demonstrated that ensemble classifiers trained on a reduced feature set from NSL-KDD could achieve up to 99.1% DDoS detection accuracy. A more recent development by [7] introduced a CNN-based DL model enhanced with attribute fusion and cross-category entropy loss, outperforming traditional classifiers in various performance metrics.

Lastly, AIS-based models have continued to show potential, particularly for early-stage detection in cloud computing environments. One such method achieved strong results across multiple evaluation criteria, reinforcing the role of bio-inspired techniques in cybersecurity research [25].

B. INFORMATION SECURITY

Information security involves a series of stages designed to enhance a system's overall security [26]. Its fundamental objective is to safeguard the three pillars of the CIA triad: Confidentiality, Integrity, and Availability. This process is typically divided into three key areas—prevention, detection, and response—each requiring careful maintenance, evaluation, and strategic planning to ensure smooth progression between stages.

In the prevention stage, organizations must establish and implement security policies, employee awareness programs, and access control mechanisms to stop attacks before they occur [27]. These elements are interconnected and should be put in place early on. Security policies are generally categorized into physical, logical, and administrative control [28]. Organizations continuously educate employees to prevent falling victim to cyberattacks. Access control provides identity verification and specific authentication and authorization levels for each user [26].

In the detection phase, network administrators and security analysts play a crucial role. One of the key technologies for intrusion detection is the IDS. IDS must be continuously improved, as even the most secure systems are susceptible to threats. IDS can detect attacks, but when a breach occurs, it alerts network administrators, who must then follow a response plan. During the response phase, organizations must be prepared to handle incidents to maintain system integrity. This is achieved by establishing an incident response strategy, which includes containment, eradication, and recovery measures.

C. INTRUSION DETECTION SYSTEMS (IDS)

One possible strategy to stop dispersed DDoS attackers from breaching secure networks is intrusion detection. Without human assistance, a successful IDS should be able to swiftly and independently detect novel DDoS attacks. Network intrusion detection systems (NIDSs) and host intrusion detection systems (HIDSs) are two subcategories of IDSs. IDS may utilize anomaly-based or signature-based techniques to detect and categorize network traffic. Training data must make sense and perform well for the anomaly-based technique, which compares network traffic flow with baseline data that has been previously established. The signature-based method, sometimes referred to as misuse detection, verifies each packet by comparing it with stored signatures or previously discovered assaults in a signature-based database. Training data is needed for anomaly-based detection, while pre-stored signatures are needed for signature-based detection. Additionally, IDSs based on signatures have a high detection rate for known intrusions but are useless in spotting unidentified threats [29].

D. CLASSIFICATION OF ATTACKS

A system's availability, confidentiality, and integrity can all be jeopardized by network intrusions. Intrusions are becoming more sophisticated, non-repetitive, and extremely covert as networks become more varied. Among the most prevalent kinds of attacks are the following:

- DDoS and DoS: Trojans are malicious programs that have the ability to intercept network communications and take over a machine in order to carry out unlawful actions. Worms are similar to Trojan horses but may replicate themselves to propagate via networks.
- User to Root (U2R): This refers to operating system-level exploits, including buffer overflow, in which a normal user grants an attacker root access.
- Backdoor: Software that is covertly installed after a website's backend or computer system has been infiltrated, permitting unwanted access even after the associated vulnerability has been fixed.

E. DISTRIBUTED DENIAL OF SERVICE (DDoS)

A DDoS attack occurs when multiple compromised systems are used simultaneously to flood a target with excessive traffic, thereby obstructing legitimate users from accessing network resources. Unlike a typical DoS attack—which originates from a single infected host—DDoS attacks leverage a network of exploited devices or virtual machines to intensify the disruption. This distributed nature makes DDoS attacks significantly more damaging and more difficult to mitigate compared to traditional DoS attacks. See Fig. 2.

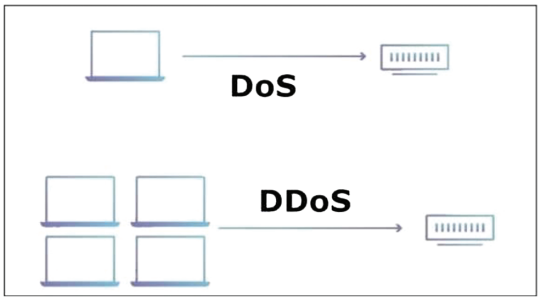

Fig. 2. DDoS attack.

It is almost impossible to fully protect against DDoS attacks due to the sheer volume of simultaneous attacks. DDoS attacks simulate normal traffic but increase dramatically, making detection and mitigation extremely challenging.

For example, in the GitHub DDoS attack of 2018, the attack peaked at 1.35 Tbps [30]. Another case in 2018 involved NETSCOUT Arbor, which suffered a 1.7 Tbps DDoS attack [31]. Similarly, Amazon Web Services (AWS) was targeted by a massive DDoS attack peaking at 2.3 Tbps [31,32]. These are among the largest DDoS attacks in recent years, causing significant financial and operational damage to industries and governments worldwide [33].

One major reason behind these attacks is the rise of internet-connected devices interacting with remote applications, which allows malicious actors to exploit and control these devices. Reference [34] stated that DDoS attacks are increasingly used because they are easy to implement, do not require high technical skills, and are supported by various platforms and applications to coordinate attacks. Attacks in DDoS use many devices (botnets), and attackers place them as "Control Servers" and can control servers and systems [35]. Attackers will send commands to servers that are rich in memory size, bandwidth, CPU power, capacity, and capabilities through many PCs online via the internet network and will play their role as server controllers. Botnet owners will not be aware of the presence of malware on their computers because they have unknowingly become part of the attack process. Reference [36] stated that DDoS attacks are carried out through proxies by attackers.

F. MACHINE LEARNING

ML technology is a machine that is developed to be able to learn by itself without direction from its user. Reference [37] describes three approaches to ML techniques: supervised learning, where this technique relies on labeled datasets to train the model, allowing the model to make accurate predictions when presented with new data; unsupervised learning models, namely techniques to explore datasets to uncover patterns or groupings without prior guidance; and semi-supervised learning models, which combine the strengths of supervised and unsupervised learning, and semi-supervised models can improve performance and accuracy while minimizing the need for extensive manual labeling.

ML techniques have the ability to detect network data flows through derived characteristics and are widely applied in IDS. Traditional techniques such as Naïve Bayes, RF, decision tree, and support vector machine have proven effective in categorizing typical and malicious activities [38,39]. However, [40] claim

that these techniques require manual features and experience difficulties when adapting to evolving attack methods or unfamiliarity. In highly complex and irregular DDoS traffic, the effectiveness of ML models will vary greatly [2]. Over time, the shortcomings of these models have encouraged a shift to DL models, which have the ability to independently identify current trends without the need for custom features.

G. DEEP LEARNING

DL, a subset of ML, distinguishes itself by its ability to learn hierarchical representations through multiple layers of nonlinear transformations. The key principle behind DL is that data consist of fundamental patterns that are interrelated in a structured hierarchy. By stacking multiple layers, DL models extract progressively complex abstractions: lower layers capture basic features like edges and gradients and higher layers detect intricate patterns such as shapes, objects, and even faces.

DL models have the ability to learn hierarchical representations of data and uncover nonlinear correlations among network features. CNNs identify spatial relationships in traffic data, while recurrent neural networks (RNNs), particularly LSTM models, can capture temporal dependencies [41,42]. Several studies have shown that CNN and LSTM models are a step ahead of traditional ML models in terms of increasing accuracy and reducing false positives [40,43]. The main challenge with DL architectures is related to the spatial and temporal dimensions of the data, which affect the ability to model the spatio-temporal complexity of attacks [44]. This development has led to the development of hybrid and ensemble deep learning architectures. These models improve detection performance by combining complementary learning mechanisms.

Hybrid and ensemble DL architectures have emerged as promising solutions to improve IDS performance. By combining multiple algorithms, these systems leverage the strengths of different models while compensating for their weaknesses. In particular, CNN–LSTM hybrids have demonstrated strong potential for capturing spatial and temporal features of network traffic [45,46].

H. DEEP NEURAL NETWORKS

Among DL architectures, deep neural networks (DNNs) are one of the most widely used frameworks for solving complex problems in fields such as cybersecurity, computer vision, and natural language processing [47]. DNNs leverage multiple interconnected layers to analyze data patterns, making them highly effective for large-scale predictive tasks.

I. PYTHON IMPLEMENTATION

Python is a powerful, high-level programming language that supports object-oriented programming. While it includes a vast ecosystem of libraries that may initially seem overwhelming, its clear and intuitive syntax makes it easy to learn and use. Developed by Guido van Rossum and officially launched in 1991, Python has grown into one of the most widely used programming languages in the field of ML and artificial intelligence.

The language offers access to an extensive suite of ML, AI, and scientific computing libraries, including—but not limited to—NumPy, SciPy, Scikit-learn, TensorFlow, Keras, and Theano [48]. For this research, model development and training were carried out using the Keras library, which operates on top of the TensorFlow framework.

J. TENSORFLOW FRAMEWORK

TensorFlow is an open-source, high-performance framework for numerical computation, particularly well-suited for DL applications. Its highly adaptable architecture allows for computations to be performed on various hardware platforms, including Tensor Processing Units (TPUs), Graphics Processing Units (GPUs), and Central Processing Units (CPUs), whether on personal computers, servers, or mobile devices. Developed by the Google Brain team in collaboration with researchers at Google AI, TensorFlow's robust computational engine supports a wide range of scientific and engineering applications [49]. It is especially advantageous for training DNNs due to its modular structure, which allows developers to configure components freely for various learning models.

K. KERAS LIBRARY

Keras is an open-source neural network library designed for ease of use and rapid development. Compatible with both TensorFlow and Theano backends, Keras enables seamless execution and model deployment. It emphasizes user-friendliness, modularity, and extensibility, making it an ideal tool for prototyping and experimenting with DL architectures. The library offers a comprehensive selection of built-in functionalities, including activation functions, normalization layers, and optimization algorithms. Its benefits include fast execution, well-documented resources, and a developer-friendly environment [50].

L. DEEP LEARNING METHOD

1). CONVOLUTIONAL NEURAL NETWORK (CNN). CNNs are very effective at recognizing fundamental patterns in data, allowing them to build more complex representations in deeper layers. CNN is a special form of multilayer neural network that uses the backpropagation algorithm, similar to many other neural network models. The CNN architecture consists of an input layer, several hidden layers, and an output layer [51]. The hidden layers include convolutional layers, pooling layers, and fully connected layer [52]. Within the convolutional layer, the input data undergoes a filtering process, where it is convolved with kernels to extract meaningful features and enhance the quality of the output [52]. Initially, random inputs and kernels are processed through the convolution operation, producing the highest output when the kernel matches a specific segment of the input. Following this, the subsampling or pooling layer serves to reduce the data's dimensionality by down-sampling the spatial dimensions of the feature maps obtained from the convolutional layer. Pooling helps decrease network complexity and the number of parameters, making computations more efficient. Among various pooling techniques, max-pooling is the most prevalent; it involves selecting the maximum value within a small pooling window. For example, setting the stride to 2 in max-pooling halves the output dimension.

The final layer type discussed is the multilayer perceptron (MLP), a type of feedforward neural network. The proposed CNN model consists of five layers and has been tested on the NSL-KDD dataset. It begins with an input layer receiving processed data, followed by a convolutional layer, then a max-pooling layer which flattens the output by a factor of two. The last layer is a fully connected one, which acts as the network's output layer.

2). LONG SHORT-TERM MEMORY (LSTM). RNNs are specifically designed to process sequential data by retaining contextual

information from previous inputs, which distinguishes them from traditional neural networks that treat each input–output pair as independent [53]. This ability to reference earlier computations during current processing steps allows RNNs to make more informed predictions, especially in tasks where temporal dependencies are critical. The "recurrent" nature of RNNs refers to this internal feedback loop that enables the model to integrate past states into ongoing decision-making.

An advanced and widely adopted variant of RNNs is the LSTM network, which has demonstrated exceptional capability in capturing long-range dependencies in sequential data [54]. LSTMs address the limitations of traditional RNNs—particularly the vanishing gradient problem—by incorporating a gating mechanism that intelligently manages the flow of information over time.

This mechanism consists of three primary components. The forget gate determines which information from the previous cell state should be discarded or retained, based on the current input and the prior hidden state. It utilizes a sigmoid activation function to assign importance values between 0 and 1, effectively filtering memory contents. Next, the input gate controls the extent to which new data is allowed to enter the cell state, again using sigmoid activation to selectively update memory with relevant incoming signals. Lastly, the output gate governs which part of the updated cell state will be exposed as output, influencing both the current prediction and the subsequent state passed to the next time step. This sophisticated gating process enables LSTMs to dynamically learn which information is essential for long-term context and which can be discarded, making them highly effective in modeling complex temporal sequences.

3). LSTM ARCHITECTURE IMPLEMENTATION. The proposed LSTM model comprises three primary layers and is applied to the NSL-KDD dataset. The framework begins with an input layer that receives the preprocessed data, followed by the core LSTM layer. This layer processes temporal dependencies and passes the result to the final fully connected output layer.

4). CLSTMNET MODEL. The CLSTMNet architecture integrates the strengths of both CNN and LSTM models, forming a hybrid seven-layer structure designed to optimize the detection of DDoS attacks. This approach utilizes the NSL-KDD dataset and aims to achieve high detection accuracy by leveraging the powerful feature extraction capabilities of CNN and the sequence learning strengths of LSTM [8].

This model architecture includes an input layer, two convolutional layers for feature extraction, followed by two max-pooling layers for down-sampling, an LSTM layer for sequence prediction, and concludes with a fully connected output layer [55]. The specific configuration of each layer, along with its respective parameters and operations, is detailed in Fig. 3.

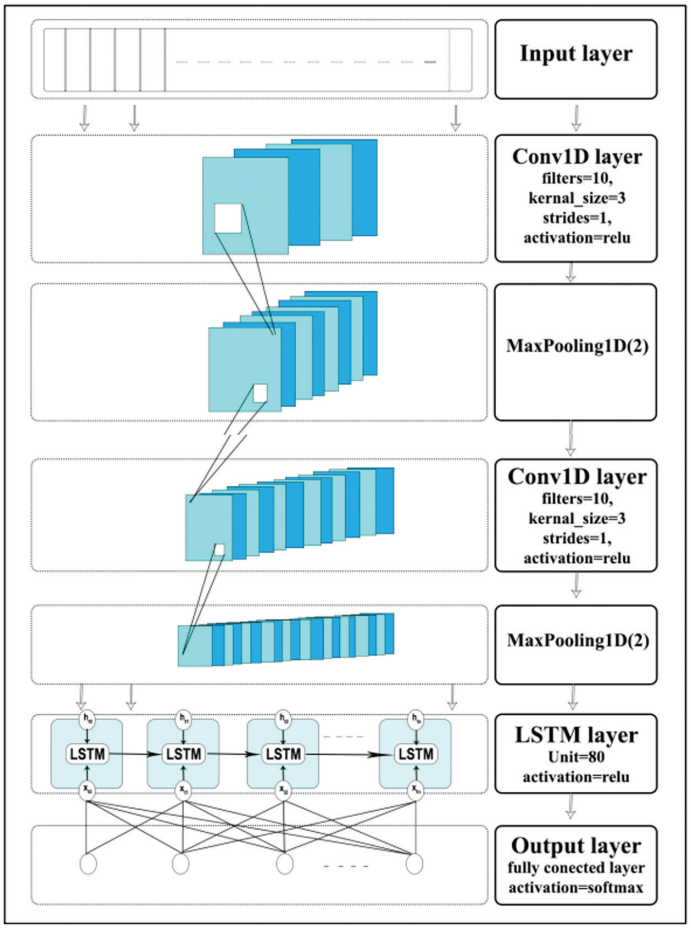

Fig. 3. The structure of CLSTMNet.

III. METHODOLOGY

This study introduces a hybrid DL model, referred to as CLSTMNet, designed specifically to enhance the detection of DDoS attacks. The model architecture integrates CNNs for automatic feature extraction and LSTM networks for learning temporal patterns within network traffic data.

The proposed system was evaluated using the NSL-KDD dataset, a widely adopted benchmark in intrusion detection research. This dataset addresses several deficiencies found in its predecessor, the KDD Cup 99, such as redundant records and imbalanced class distributions, making it more suitable for modern ML evaluation. The dataset includes four attack categories—DoS, Probe, Remote-to-Local (R2L), and User-to-Root (U2R)—with 41 input features and labeled outputs [56].

To prepare the data for training, several preprocessing steps were applied. First, categorical attributes were encoded using one-hot encoding to enable compatibility with the neural network model. Then, numerical values were normalized to a fixed range to stabilize training and accelerate convergence. The final dataset was randomly shuffled and split into training and testing subsets to ensure generalizability and avoid overfitting.

The CLSTMNet model consists of a total of seven layers. It begins with two convolutional layers that extract spatial features from the input. These are followed by a max pooling layer to reduce dimensionality and computational cost. The output is then passed to two stacked LSTM layers, which learn temporal dependencies in the sequence of network activities. Finally, a fully connected dense layer with a softmax activation function is used for classification. The model was implemented using TensorFlow and Keras frameworks. See Fig. 4.

Model training was carried out using categorical cross-entropy as the loss function and Adam as the optimizer. The training process involved 100 epochs and a batch size of 64, with performance monitored using accuracy and loss metrics. To prevent overfitting, dropout layers and early stopping mechanisms were included. Evaluation was performed on the testing set using several metrics, including accuracy, precision, recall, and F1-score, to assess both overall and class-specific performance.

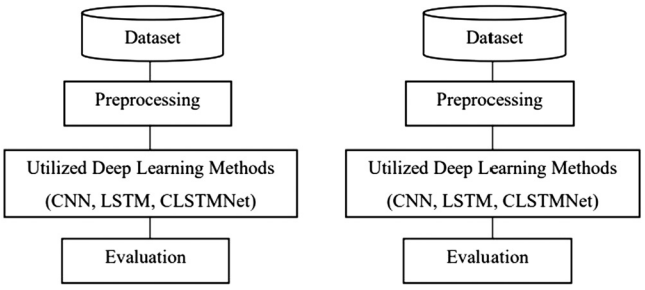

Fig. 4. Research methodology model.

IV. RESULTS AND DISCUSSION

To assess the performance of the proposed CLSTMNet model, the evaluation was conducted using the testing subset of the NSL-KDD dataset. The assessment employed four commonly used classification metrics: accuracy, precision, recall, and F1-score. Based on the results, CLSTMNet demonstrated an impressive classification accuracy of 98.23%, indicating its robustness in reliably identifying DDoS attacks. As illustrated in Table I, the model's performance was benchmarked against several baseline classifiers, including conventional ML approaches such as RF and decision tree, as well as DL models like CNN, LSTM, and GRU. The comparative results clearly show that CLSTMNet consistently outperforms these models in terms of both accuracy and detection effectiveness.

These results indicate that CLSTMNet outperforms both classical ML models and other DL approaches across all evaluation metrics. The superior performance can be attributed to the combination of convolutional layers, which capture spatial dependencies in the input features, and LSTM layers, which model sequential and temporal patterns in network traffic data.

In terms of precision, CLSTMNet achieves a score of 98.05%, suggesting a low rate of false positives, which is critical in IDSs to avoid unnecessary alerts. The recall score of 98.10% indicates that the model is highly effective in identifying actual attack instances, minimizing the number of undetected intrusions. A high F1-score (98.07%) further confirms that the model maintains a strong balance between precision and recall.

Figure 3 illustrates the accuracy and loss trends over 100 training epochs. The accuracy curve shows a consistent increase and early convergence, while the loss steadily decreases, confirming the model's stable learning behavior.

Moreover, the proposed model was evaluated not only on the overall dataset but also on specific attack types. CLSTMNet showed consistent detection performance across DoS, Probe, R2L, and U2R categories. This generalizability highlights the robustness of the architecture in dealing with both frequent and infrequent attack patterns.

These findings reinforce the potential of hybrid DL models for network security applications. Compared to traditional systems, CLSTMNet offers better accuracy, reduced false alarms, and higher detection rates—all of which are essential for real-time DDoS mitigation.

Table I. Performance comparison of CLSTMNet and other models

Model	Accuracy	Precision	Recall	F1-score
Decision tree (DT)	91.76%	91.24%	91.40%	91.32%
Random forest (RF)	93.25%	93.10%	93.20%	93.15%
CNN	94.85%	94.62%	94.71%	94.66%
LSTM	95.42%	95.17%	95.30%	95.23%
GRU	96.20%	95.92%	96.00%	95.96%
CLSTMNet (proposed)	**98.23%**	**98.05%**	**98.10%**	**98.07%**

V. CONCLUSIONS

This study presented CLSTMNet, an innovative hybrid DL model that combined the strengths of CNN and LSTM networks to address the critical issue of DDoS detection. Within this architecture, CNN acted as an efficient feature extractor, while LSTM leverages its temporal memory capabilities to support sequential prediction. The CLSTMNet framework comprised seven layers, each of which was tailored to enhance the model's ability to capture and classify DDoS patterns effectively.

Experiments conducted on the NSL-KDD dataset demonstrated that CLSTMNet consistently outperformed both standalone CNN and LSTM models, achieving notable results across key evaluation metrics—accuracy, precision, recall, and F1-score. Trained over five independent sessions using Python and TensorFlow, the model achieved peak accuracy of 99.20%, confirming its robustness and reliability for DDoS detection tasks.

Beyond its strong empirical performance, the CLSTMNet model showed promise for broader cybersecurity applications, including the identification of diverse attack types. Future research directions include validating the model on additional benchmark datasets to assess generalization, exploring parallel architectural modifications, integrating ensemble techniques such as majority voting, and incorporating attention mechanisms to enhance detection of underrepresented classes. Furthermore, deploying CLSTMNet in real-time environments could open pathways for practical implementation in dynamic network security systems.

CONFLICT OF INTEREST STATEMENT

The authors declare that they have no conflicts of interest to report regarding the present study.

REFERENCES

[1] A. Petrosyan, "Global number of internet users 2005-2024," Statista. [Online]. 2024. Available: https://www.statista.com/statistics/273018/number-of-internet-users-worldwide/, Accessed on : Apr. 17, 2025

[2] H. Liu and B. Lang, "Machine learning and deep learning methods for intrusion detection systems: A survey," *Appl. Sci.*, vol. 9, no. 20, p. 4396, Oct. 2019, DOI: https://doi.org/10.3390/app9204396.

[3] A. Banitalebi Dehkordi, M. Soltanaghaei, and F. Z. Boroujeni, "The DDoS attacks detection through machine learning and statistical methods in SDN," *J. Supercomput.*, vol. 77, no. 3, pp. 2383–2415, Mar. 2021, DOI: https://doi.org/10.1007/s11227-020-03323-w.

[4] N. Bindra and M. Sood, "Detecting DDoS attacks using machine learning techniques and contemporary intrusion detection dataset," *Autom. Control Comput. Sci.*, vol. 53, no. 5, pp. 419–428, Sep. 2019, DOI: https://doi.org/10.3103/S0146411619050043.

[5] Y. Otoum, D. Liu, and A. Nayak, "DL-IDS: A deep learning–based intrusion detection framework for securing IoT," *Trans. Emerg. Telecommun. Technol.*, vol. 33, no. 3, p. e3803, Mar. 2022, DOI: https://doi.org/10.1002/ett.3803.

[6] X. Yuan *et al.*, "Adversarial examples: Attacks and defenses for deep learning," 2017.

[7] L. Ma *et al.*, "A deep learning-based DDoS detection framework for Internet Of Things," in *ICC 2020-2020 IEEE International Conference on Communications (ICC)*, IEEE, pp. 1–6, Jun. 2020, DOI: https://doi.org/10.1109/ICC40277.2020.9148944.

[8] A. Tasdelen and B. Sen, "A hybrid CNN-LSTM model for pre-miRNA classification," *Sci. Rep.*, vol. 11, no. 1, p. 14125, Jul. 2021, DOI: https://doi.org/10.1038/s41598-021-93656-0.

[9] J. Donahue *et al.*, "Long-term recurrent convolutional networks for visual recognition and description," *2015 IEEE Conf. Comput. Vision Pattern Recogn. (CVPR)*, IEEE, vol. 2015, pp. 2625–2634, Jun. 2015, DOI: https://doi.org/10.1109/CVPR.2015.7298878.

[10] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," Apr. 2015.

[11] T. D. Diwan, S. Choubey, and H. S. Hota, "A detailed analysis on NSL-KDD dataset using various machine learning techniques for intrusion detection," *Turk. J. Comput. Math. Educ.*, vol. 12, no. 11, pp. 2954–2968, May 2021.

[12] A. S. Boroujerdi and S. Ayat, "A robust ensemble of neuro-fuzzy classifiers for DDoS attack detection," *Proc. 2013 3rd Int. Conf. Comput. Sci. Netw. Technol.*, IEEE, pp. 484–487, Oct. 2013, DOI: https://doi.org/10.1109/ICCSNT.2013.6967159.

[13] G. S. Kushwah and S. T. Ali, "Detecting DDoS attacks in cloud computing using ANN and black hole optimization," in *2017 2nd International Conference on Telecommunication and Networks (TEL-NET)*, IEEE, pp. 1–5, Aug. 2017, DOI: https://doi.org/10.1109/TEL-NET.2017.8343555.

[14] O. Igbe, O. Ajayi, and T. Saadawi, "Denial of service attack detection using dendritic cell algorithm," in *2017 IEEE 8th Annual Ubiquitous Computing, Electronics and Mobile Communication Conference (UEMCON)*, IEEE, pp. 294–299, Oct. 2017, DOI: https://doi.org/10.1109/UEMCON.2017.8249054.

[15] S. Sharma *et al.*, "Analysis of NSL KDD dataset using classification algorithms for intrusion detection system," *Recent Pat. Eng.*, vol. 13, no. 2, pp. 142–147, May 2019, DOI: https://doi.org/10.2174/1872212112666180402122150.

[16] A. R. Yusof *et al.*, "Adaptive feature selection for denial of services (DoS) attack," in *2017 IEEE Conference on Application, Information and Network Security (AINS)*, IEEE, pp. 81–84, Nov. 2017, DOI: https://doi.org/10.1109/AINS.2017.8270429.

[17] M. Idhammad, K. Afdel, and M. Belouch, "Detection system of HTTP DDoS attacks in a cloud environment based on information theoretic entropy and random forest," *Secur. Commun. Netw.*, vol. 2018, pp. 1–13, Jun. 2018, DOI: https://doi.org/10.1155/2018/1263123.

[18] M. Idhammad, K. Afdel, and M. Belouch, "Semi-supervised machine learning approach for DDoS detection," *Appl. Intell.*, vol. 48, no. 10, pp. 3193–3208, Oct. 2018, DOI: https://doi.org/10.1007/s10489-018-1141-2.

[19] A. M. Derakhsh, P. Daneshjoo, and C. Delara, "Using genetic algorithm to improve Bernoulli Naïve Bayes Algorithm in order to detect DDoS attacks in cloud computing platform," *Int. J. Sci. Eng. Investig.*, vol. 7, no. 22, pp. 1–7, Jan. 2018.

[20] K. S. Hoon *et al.*, "Critical review of machine learning approaches to apply big data analytics in DDoS forensics," in *2018 International Conference on Computer Communication and Informatics (ICCCI)*, IEEE, pp. 1–5, Jan. 2018, DOI: https://doi.org/10.1109/ICCCI.2018.8441286.

[21] F. Mukhametzyanov *et al.*, "The neural network model of DDoS attacks identification for information management," *Int. J. Supply Chain Manag.*, vol. 8, no. 5, pp. 1–10, 2019.

[22] P. Verma, S. Tapaswi, and W. W. Godfrey, "An adaptive threshold-based attribute selection to classify requests under DDoS attack in cloud-based systems," *Arab. J. Sci. Eng.*, vol. 45, no. 4, pp. 2813–2834, Apr. 2020, DOI: https://doi.org/10.1007/s13369-019-04178-x.

[23] S. Hosseini and M. Azizi, "The hybrid technique for DDoS detection with supervised learning algorithms," *Comput. Netw.*, vol. 158, pp. 35–45, Jul. 2019, DOI: https://doi.org/10.1016/j.comnet.2019.04.027.

[24] S. Das *et al.*, "DDoS intrusion detection through machine learning ensemble," in *2019 IEEE 19th International Conference on Software Quality, Reliability and Security Companion (QRS-C)*, IEEE, pp. 471–477, Jul. 2019, DOI: https://doi.org/10.1109/QRS-C.2019.00090.

[25] D. J. Prathyusha and G. Kannayaram, "A cognitive mechanism for mitigating DDoS attacks using the artificial immune system in a cloud environment," *Evol. Intell.*, vol. 14, no. 2, pp. 607–618, Jun. 2021, DOI: https://doi.org/10.1007/s12065-019-00340-4.

[26] J. LaPiedra, "The information security process prevention, detection and response," Global Information Assurance Certification Paper. [Online]. 2018. Available: https://www.giac.org/paper/gsec/501/information-security-process-prevention-detection-response/101197, Accessed on: Apr. 17, 2025

[27] M. Chapple and D. Seidl,*CompTIA CySA+ Study Guide: Exam CS0-001*. Indianapolis, Indiana: John Wiley & Sons, Inc, 2017, [Online]. Available: https://dl.hellodigi.ir/dl.hellodigi.ir/dl/book/CompTIA Cybersecurity Analyst %28CSA%2B%29 Study Guide Exam CS0-001.pdf, Accessed on: Dec. 15, 2024.

[28] A. Özalp, Z. Albayrak, and A. Zengin, "Expansion of wireless networks using IEEE 802.3af protocol in protected areas," *Conf. 5th Int. Symp. Innov. Technol. Eng. Sci.*, 2017.

[29] A. Avalappampatty Sivasamy and B. Sundan, "A Dynamic Intrusion Detection System Based on Multivariate Hotelling's T ^{2} Statistics Approach for Network Environments," *Sci. World J.*, vol. 2015, no. 1, Jan. 2015, DOI: https://doi.org/10.1155/2015/850153.

[30] Cloudflare, "What is a denial-of-service (DoS) attack?" Cloudfilare. [Online]. 2024. Available: https://www.cloudflare.com/ru-ru/learning/ddos/glossary/denial-of-service/, Accessed: Mar. 17, 2025.

[31] C. Cimpanu, "AWS said it mitigated a 2.3 Tbps DDoS attack, the largest ever," ZDNet. [Online]. 2020. Available: https://www.zdnet.com/article/aws-said-it-mitigated-a-2-3-tbps-ddos-attack-the-largest-ever/, Accessed on: Mar. 17, 2025.

[32] Cloudflare, "Famous DDoS attacks: The largest DDoS attacks of all time," Cloudflare.

[33] D. Anstee *et al.*, "Worldwide infrastructure security report," Arbor Network Special Report Volume XI. [Online]. 2018. Available: https://kapost-files-prod.s3.amazonaws.com/published/569e85ff426d9e582400000a/wisr-report.pdf, Accessed on: Mar. 17, 2025.

[34] F. S. de Lima Filho *et al.*, "Smart detection: An online approach for DoS/DDoS attack detection using machine learning," *Secur. Commun. Netw.*, vol. 2019, pp. 1–15, Oct. 2019, DOI: https://doi.org/10.1155/2019/1574749.

[35] T. A. Tuan *et al.*, "Performance evaluation of Botnet DDoS attack detection using machine learning," *Evol. Intell.*, vol. 13, no. 2, pp. 283–294, Jun. 2020, DOI: https://doi.org/10.1007/s12065-019-00310-w.

[36] H. Beitollahi and G. Deconinck, "ConnectionScore: A statistical technique to resist application-layer DDoS attacks," *J. Ambient Intell. Humaniz. Comput.*, vol. 5, no. 3, pp. 425–442, Jun. 2014, DOI: https://doi.org/10.1007/s12652-013-0196-5.

[37] M. Alabadi and Z. Albayrak, "Q-Learning for securing cyber-physical systems : A survey," *2020 Int. Congr. Hum.-Comput. Interaction,*

Optim. Rob. Appl. (HORA), IEEE, vol. 2020, pp. 1–13, Jun. 2020, DOI: https://doi.org/10.1109/HORA49412.2020.9152841.

[38] A. Khraisat *et al.*, "Survey of intrusion detection systems: Techniques, datasets and challenges," *Cybersecur.*, vol. 2, no. 1, p. 20, Dec. 2019, DOI: https://doi.org/10.1186/s42400-019-0038-7.

[39] N. Shone *et al.*, "A deep learning approach to network intrusion detection," *IEEE Trans. Emerg. Top. Comput. Intell.*, vol. 2, no. 1, pp. 41–50, Feb. 2018, DOI: https://doi.org/10.1109/TETCI.2017.2772792.

[40] G. Karatas, O. Demir, and O. K. Sahingoz, "Increasing the performance of machine learning-based IDSs on an imbalanced and up-to-date dataset," *IEEE Access*, vol. 8, pp. 32150–32162, 2020, DOI: https://doi.org/10.1109/ACCESS.2020.2973219.

[41] R. Doriguzzi-Corin *et al.*, "Lucid: A practical, lightweight deep learning solution for DDoS attack detection," *IEEE Trans. Netw. Serv. Manag.*, vol. 17, no. 2, pp. 876–889, Jun. 2020, DOI: https://doi.org/10.1109/TNSM.2020.2971776.

[42] Q. Niyaz, W. Sun, and A. Y. Javaid, "A deep learning based DDoS detection system in software-defined networking (SDN)," *ICST Trans. Secur. Saf.*, vol. 4, no. 12, p. 153515, Dec. 2017, DOI: https://doi.org/10.4108/eai.28-12-2017.153515.

[43] J. Lansky *et al.*, "Deep learning-based intrusion detection systems: A systematic review," *IEEE Access*, vol. 9, pp. 101574–101599, 2021, DOI: https://doi.org/10.1109/ACCESS.2021.3097247.

[44] V. Rasikha and P. Marikkannu, "An ensemble deep learning-based cyber attack detection system using optimization strategy," *Knowl. Based Syst.*, vol. 301, p. 112211, Oct. 2024, DOI: https://doi.org/10.1016/j.knosys.2024.112211.

[45] H. C. Altunay and Z. Albayrak, "A hybrid CNN+LSTM-based intrusion detection system for industrial IoT networks," *Eng. Sci. Technol. an Int. J.*, vol. 38, p. 101322, Feb. 2023, DOI: https://doi.org/10.1016/j.jestch.2022.101322.

[46] C. Yin *et al.*, "A deep learning approach for intrusion detection using recurrent neural networks," *IEEE Access*, vol. 5, pp. 21954–21961, 2017, DOI: https://doi.org/10.1109/ACCESS.2017.2762418.

[47] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015, DOI: https://doi.org/10.1038/nature14539.

[48] F. Pedregosa *et al.*, "Scikit-learn: Machine Learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, 2011.

[49] L. Rampasek and A. Goldenberg, "TensorFlow: Biology's gateway to deep learning?," *Cell Syst.*, vol. 2, no. 1, pp. 12–14, Jan. 2016, DOI: https://doi.org/10.1016/j.cels.2016.01.009.

[50] B. J. Erickson *et al.*, "Toolkits and libraries for deep learning," *J. Digit. Imaging*, vol. 30, no. 4, pp. 400–405, Aug. 2017, DOI: https://doi.org/10.1007/s10278-017-9965-6.

[51] H. C. Altunay *et al.*, "Analysis of anomaly detection approaches performed through deep learning methods in SCADA systems," in *2021 3rd Int. Congr. Hum. Comput. Interaction, Optim. Rob. Appl. (HORA)*, IEEE, vol. 2021, pp. 1–6, Jun. 2021, DOI: https://doi.org/10.1109/HORA52670.2021.9461273.

[52] H. Zeng *et al.*, "Convolutional neural network architectures for predicting DNA–protein binding," *Bioinformatics*, vol. 32, no. 12, pp. i121–i127, Jun. 2016, DOI: https://doi.org/10.1093/bioinformatics/btw255.

[53] P. Gudikandula, "Recurrent neural networks and LSTM explained," Medium. [Online]. 2021. Available: https://purnasaigudikandula.medium.com/recurrent-neural-networks-and-lstm-explained-7f51c7f6bbb9, Accessed on: Mar. 19, 2025

[54] D. Thakur, "LSTM and its equations," Medium. [Online]. 2021. Available: https://medium.com/@divyanshu132/lstm-and-its-equations-5ee9246d04af, Accessed on: Mar. 18, 2025.

[55] A. S. Ahmed ISSA and Z. ALBAYRAK, "CLSTMNet: A deep learning model for intrusion detection," *J. Phys. Conf. Ser.*, vol. 1973, no. 1, p. 012244, Aug. 2021, DOI: https://doi.org/10.1088/1742-6596/1973/1/012244.

[56] M. Tavallaee *et al.*, "A detailed analysis of the KDD CUP 99 data set," in *2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications*, IEEE, pp. 1–6, Jul. 2009, DOI: https://doi.org/10.1109/CISDA.2009.5356528.