International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 06 | June - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

SDN Attack Identification Model Using Hybrid CNN – LSTM with Attention Mechanism

P. Loganayagi[1], Dr. G. Ramesh[2]

[1]*Information Technology, K.L.N College of Engineering*
[2]*Information Technology, K.L.N College of Engineering*

Abstract - SDN provides centralised control and programmability, but because of its open and centralised architecture, it is extremely susceptible to cyberattacks like Distributed Denial of Service (DDoS), infiltration, and botnets. In terms of accuracy and flexibility, traditional intrusion detection systems frequently fall short of the changing requirements of SDN settings. In order to solve this, we suggest a hybrid deep learning model that incorporates Long Short-Term Memory (LSTM) networks and Convolutional Neural Networks (CNN), augmented with an Attention mechanism. In order to increase accuracy and interpretability, CNN layers take out spatial information from traffic data, LSTM layers record temporal dependencies, and the Attention mechanism highlights important elements. The CICIDS 2017 dataset is used to train and assess the model, utilising pre - processing methods such as class balancing, label encoding, and normalisation. According to experimental results, our model outperforms conventional models such standalone CNNs and statistical techniques, achieving an accuracy of 93.43%. It performs admirably in a variety of attack scenarios, such as DDoS, probe, and penetration. This study establishes the foundation for real-time, scalable deployment and demonstrates the potential of hybrid deep learning models in SDN cybersecurity. Future research will concentrate on improving the detection of zero-day attacks and tailoring the model for edge computing settings with TensorFlow Lite.

Key Words: SDN Security, Intrusion Detection, CNN-LSTM Hybrid, Attention Mechanism, Cyberattack Detection.

1.INTRODUCTION

A logically centralised SDN controller offers centralised network control by separating the control plane from the data plane. Software-Defined Networking (SDN) is a revolutionary networking paradigm. Network managers may now dynamically modify routing policies, programmatically control traffic flows, and granularly enforce security configurations thanks to this architectural change. By using programmable interfaces, SDN streamlines operations, increases network flexibility, and fosters quick innovation.

The controller functions as the "brain" of the infrastructure in an SDN-enabled network, deciding crucially how data packets should be routed via switches and routers. By employing southbound protocols like OpenFlow to receive instructions from the controller, these devices transform into basic forwarding components. In the meantime, administrators and external apps can communicate with the controller using northbound APIs to make wise decisions.
Despite its benefits, this centralisation also creates a serious weakness. Cyber adversaries see the SDN controller to be a high-value target and a single point of failure. The necessity for strong, intelligent security methods included into the SDN architecture is highlighted by the possibility that the entire network might be interfered with or altered if compromised.

Although SDN's dynamic and customisable nature increases the attack surface, it also creates new opportunities for network management. SDN's programmable flows, controller logic, and open interfaces can be used by cyber attackers to initiate a variety of complex assaults. The main attack types that pose a threat to SDN environments are listed below:

A. Denial of Service (DoS) and Distributed Denial of Service (DDoS)
These attacks send a huge amount of unauthorised traffic or connection requests to the SDN controller or network devices. The SDN controller may get overloaded since it has to handle every request, which could result in significant performance deterioration, higher latency, or total service outages. Because SDN has centralised control, a single compromised point can take down the entire network, making these attacks very harmful.

B. Brute Force
Brute force attacks include adversaries trying every possible combination of credentials in an effort to obtain unauthorised access to SDN components. Once they have access, attackers can alter current configurations or introduce malicious flows, giving them unapproved control over traffic flow and possibly even the ability to exfiltrate data.

C. Botnets
A network of compromised endpoints (bots) under the control of a remote attacker makes up a botnet. Botnets are used to plan extensive, dispersed attacks on the controller or edge nodes in an SDN environment. These attacks are difficult to stop with static or rule-based systems because they frequently change their IP addresses and traffic patterns to avoid detection.

E. Web and API
RESTful APIs and web interfaces for management and monitoring are frequently exposed by contemporary SDN controllers. SQL injection, Cross-Site Scripting (XSS), Cross-Site Request Forgery (CSRF), and command injection are some of the methods that attackers use to take advantage of these APIs. Complete control of the network and full administrative access could result from a successful web-based attack.

2. RELATED WORK

In previous studies, the use of deep learning models for intrusion detection in SDN systems was examined. In order to detect DDoS attacks, for instance, a hybrid CNN-LSTM model demonstrated

high accuracy by collecting both the temporal and spatial aspects of network traffic. An attention-based CNN-LSTM architecture was introduced in another study, which improved performance in identifying various types of attacks by emphasising significant features in the data. These studies show how hybrid models can enhance network security, especially when they incorporate attention approaches. SDN technology has been applied extensively across a wide range of industries and can boost network capacity. Ouamari MA et al. identified issues with the data exchange between the corporate office and local branches via the wide area network. To solve this problem, they proposed the SDN-WAN technique. First, modify network management to meet service requirements. The twin optimisation challenge of average request latency and survival was then used to fix the server's latency problem. The results showed that, in comparison to traditional methods, the research strategy significantly improved the system's performance [7]. For the regular operation of the network, Garg et al. created an SDN-based framework that streamlines network management and enhances network communication [9].

In recent years, the CNN algorithm has been widely applied across numerous industries with promising results. Chen et al. proposed a unique paradigm to accelerate computations for gas detection. CNN-based Memristor is combined in this model. The study claims that the model is built using convolution cores of various sizes, employs a memristor to increase the hardware structure's overall utilisation rate for faster operation, and uses the multi-dimensional convolution approach to extract feature information of various dimensions [13]. In their radar study, Bao and Yang found that the signal becomes unstable due to human mobility and other factors. Consequently, a new method of personnel counting was developed based on CNN. In order to complete the counting task, researchers use this technique to try to obtain more clear graphical data. The results demonstrated the effectiveness of this approach and yielded the intended result [14].

Chen et al. developed a model of laser-induced breakdown spectroscopy in combination with a two-dimensional CNN algorithm for the field of rock investigation because there are currently few simultaneous multi-task models available. This model is constructed using two distinct output types to enable the simultaneous execution of classification and recycling tasks. The results showed that this model functioned more accurately than traditional models [15]. Xue et al. developed an architecture search technique to address the issue of CNN architecture design, which necessitates a substantial amount of human and computational resources. This method adapts the strategy based on adaptive mutation neural structure to automate CNN architecture design. The results showed that this method achieved the intended results, reduced calculation time, and saved labour cost .[17]. By facilitating the development of independent verification and Tamper-resistance mechanisms, Khalid et al. provided a means of ensuring the security of the Internet of Things through SDN [19].

The usefulness of hybrid deep learning models, especially CNN-LSTM architectures with attention mechanisms, for Software-Defined Networking (SDN) intrusion detection has been shown in recent studies. These models improve the accuracy of identifying threats like DDoS and other cyberattacks by capturing both temporal and spatial trends in network traffic.

SDN has also been used in real-world situations in a number of research. For instance, Garg et al. created an SDN framework to simplify network administration, whereas Ouamari et al. employed SDN-WAN to maximise data interchange across wide-area networks. CNN algorithms have also been widely used in a variety of fields. For example, Chen et al. used CNN in conjunction with memristors to identify gases quickly, Bao and Yang used CNN in radar to accurately count people, and other researchers used CNN in conjunction with spectroscopy to classify and recycle rocks. Furthermore, Khalid et al. used SDN to improve IoT security with tamper-resistance characteristics, and Xue et al. presented an automatic CNN architecture search method to lessen human work. All things considered, these studies show how CNN and SDN technologies can be applied widely across sectors, improving security, accuracy, and efficiency in a range of intelligent systems.

3. OPTIMIZATION METHOD FOR SDN ATTACK IDENTIFICATION BASED ON HYBRID CNN-LSTM WITH ATTENTION MECHANISM

A. SDN ATTACK IDENTIFICATION MODEL CONSTRUCTION

The suggested model combines the advantages of Convolutional Neural Networks (CNN), Long Short-Term Memory (LSTM) networks, and an Attention Mechanism into a single, hybrid deep learning framework in order to identify and categorise different cyberattacks in Software-Defined Networking (SDN) environments. Although SDN offers dynamic network programmability and centralised control, its architecture also poses serious security risks, including susceptibility to brute-force assaults, infiltration efforts, and Distributed Denial of Service (DDoS) attacks. Conventional intrusion detection systems, which are generally based on static rules or predetermined signatures, are not very flexible in real-time traffic settings and frequently fail to recognise changing or new assault patterns.

To fill in these shortcomings, the CNN-LSTM-Attention hybrid model is suggested. In order to efficiently capture static patterns and local correlations in the data, CNN layers are used to automatically extract spatial characteristics from raw network traffic data. After that, these features are sent through LSTM layers, which are skilled in simulating sequential behaviour and temporal dependencies in traffic flows, which is an essential skill for spotting time-dependent attacks. The Attention Mechanism, which enables the system to dynamically prioritise the most pertinent elements in each input sequence, is included to further narrow the model's focus. By highlighting the characteristics that most influence the detection result, this greatly improves the model's performance and interpretability.

The CICIDS 2017 dataset, which includes actual normal and malicious traffic data, is used to train and evaluate the model. The dataset is prepared for training through extensive pre-processing procedures like normalisation, label encoding, and class balancing. The experimental findings show that the suggested hybrid model detects and classifies a variety of attacks with exceptional accuracy—exceeding 93.43%. In addition to increasing prediction accuracy, the attention layer allows for scalable deployment in real-time SDN contexts. An important development in intelligent SDN security is represented by this concept.

Convolution Neural Network (CNN) Layers

The suggested model's Convolutional Neural Network (CNN) component is in charge of gleaning spatial characteristics from SDN traffic data. These characteristics include byte count, packet size, flow time, and additional local dependencies in the input sequences. CNNs are very good at spotting spatially linked patterns, such port scans, burst traffic, and steady flow characteristics that point to malicious activities.
Since network traffic data is handled as a time series, 1D convolutional layers are used in the model. To find local patterns in the input data, a kernel (or filter) of size three moves over it. A Rectified Linear Unit (ReLU) activation function, which adds non-linearity and aids in learning complicated features, comes after each convolution process. After that, the output is run through a Max-Pooling layer, which increases the computational efficiency of the model by reducing the dimensionality while keeping the most noticeable features.

1D Convolution Operation:

$$f_i^{(l)} = \sum_{j=1}^{k} \omega_j^{(l)} . x_{i+j-1} + b^{(l)}$$

Max Pooling:

$$p_i = max(x_i, x_{i+1}, \dots \dots, x_{i+k-1})$$

Long Short – Term Memory (LSTM) layers

The suggested hybrid model's Long Short-Term Memory (LSTM) component is made to identify sequential patterns and temporal dependencies in network traffic data. This is essential for identifying time-dependent behaviours that develop across a sequence of data flows rather than in solitary packets, including slow-paced attacks, infiltration efforts, or low-rate DDoS attacks. One or two LSTM layers, each with 64–128 memory units, are included in the model. By processing the sequence of spatial information that the CNN collected, these layers are able to remember previous traffic states and determine how current traffic is related to previous behaviours. Dropout regularisation, which randomly disables a portion of LSTM units during training, is used to lower the danger of overfitting.

LSTM networks can use gated techniques to learn long-term dependencies. The following formulas can be used to explain how an LSTM cell functions internally:

$$\mathbf{f_t} = \boldsymbol{\sigma}(W_f . [\mathbf{h_{t-1}}, \mathbf{x_t}] + \mathbf{b_f})$$ (Forget gate)

$$i_t = \sigma(W_i . [h_{t-1}, x_t] + b_i)$$ (Input gate)

$$\bar{C}_t = tanh(W_C . [h_{t-1}, x_t] + b_c)$$ (Cell input)

$$C_t = f_t . C_{t-1} + i_t . \bar{C}_t$$ (Cell State update)

$$\sigma_t = \sigma(W_0 [h_{t-1}, x_t] + b_0)$$ (Output gate)

$$h_t = O_t . tanh (C_t)$$ (Hidden State Output)

Attention Layers

In order to improve the hybrid CNN-LSTM model's performance and interpretability for SDN attack detection, the Attention Layer is essential. The attention mechanism decides which portions of the LSTM outputs are crucial for the final classification once the CNN has extracted spatial information and the LSTM has recorded temporal dependencies.

Each hidden state in the LSTM output sequence is given a weight by the attention mechanism, which then computes a context vector. These weights show how pertinent the data from each time step is to the current forecast. As a result, the model learns to "focus" on the most important flow characteristics, like odd timing gaps, sudden spikes, or repeating behaviours — all of which are typical signs of malicious activity.

The process involves in three main steps

1. $Score(h_t) = tanh\ (W_a h_t + b_a)$
2. $\alpha_t = \frac{\exp(Score(h_t))}{\sum_{i=1}^{T} \exp\ (Score(h_i))}$
3. $C = \Sigma_{t=1}^{T}\ \alpha_t h_t$

Final Dense Output

$$\hat{y} = softmax(w_{d^c} + b_d)$$

Where,

- h_t: hidden state from LSTM at time t
- α_t: attention weight
- **C**: context vector used for prediction
- $\hat{y}$: output classification vector

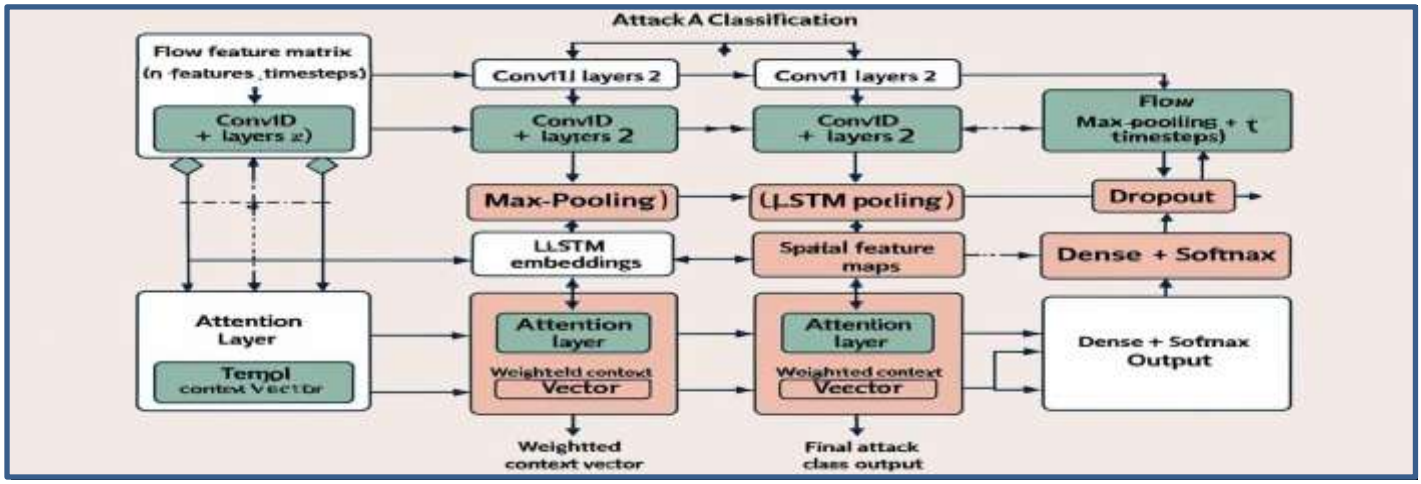


FIGURE 1. Hybrid CNN-LSTM Model

Convolutional neural networks (CNNs), long short-term memory (LSTM) networks, and attention mechanisms are all included in this hybrid deep learning architecture, which is intended for network attack categorization. This system's main input is a flow feature matrix, which comprises n features and temporal data across t time steps. These features usually depict network traffic attributes like packet sizes, time intervals, or protocol kinds. This model's objective is to correctly categorize various cyberattack kinds or identify whether the flow is benign.

The input flow feature matrix is first fed into three parallel branches of multiple Conv1D layers. By sliding filters over the time dimension, each Conv1D layer learns spatial-temporal correlations and is in charge of identifying localized temporal patterns in the input data. To better capture temporal and spatial dependencies, these convolutional outputs are further fed into other processing pathways.

A Max Pooling layer, which lowers the dimensionality while maintaining the most noticeable features, is applied to the Conv1D features in the first branch. An LSTM layer, which can describe sequential dependencies and preserve context across time steps, receives the pooled features after that. The Attention layer receives the sequence of embeddings from the LSTM and utilizes a weighted context vector to preferentially focus on the most significant time steps. After a SoftMax activation and a Dense layer, this vector—which highlights the most pertinent portions of the sequence—produces a probabilistic output that identifies the attack class.

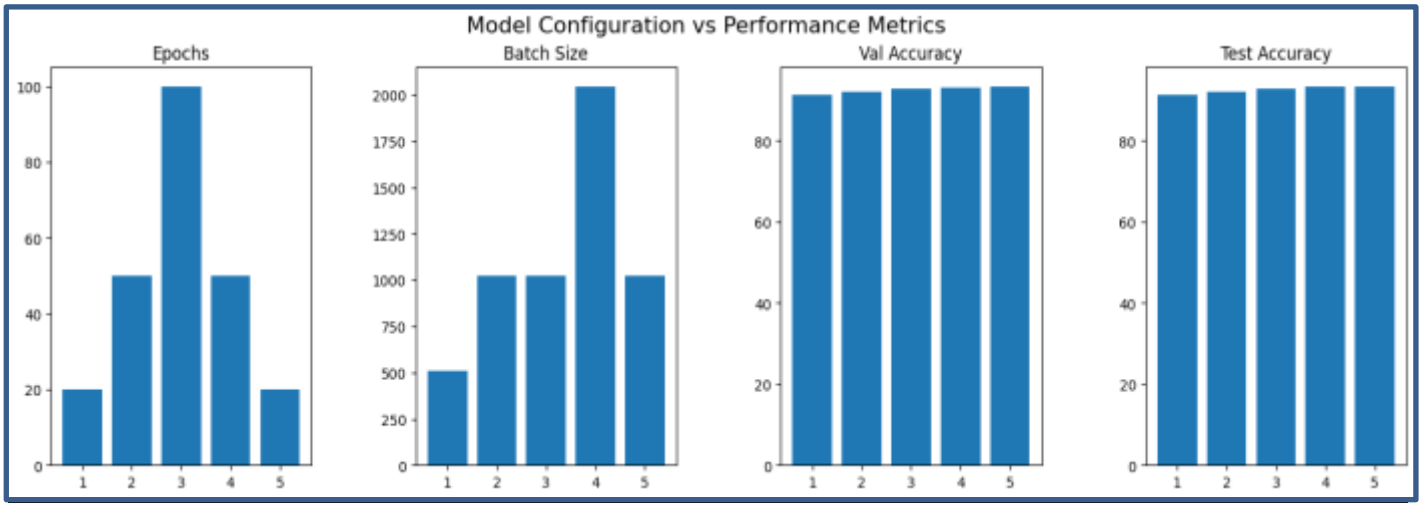
The third branch is easier to understand. Dropout, which helps regularize the model by randomly deactivating neurons during training to minimize overfitting, comes after Max Pooling over the flow matrix along the time dimension. The output serves as a quicker, easier choice pathway by being sent straight into a Dense + SoftMax layer for categorization. Following the production of their respective dense predictions or weighted context vectors by the parallel branches, these outputs are fused together, and the resulting features are then fed into a final Dense + SoftMax layer. A more reliable final prediction is achieved by integrating many feature representations, some of which are centred on temporal sequences and others on spatial patterns, using this fusion technique.

Essentially, this architecture builds a high-performance, end-to-end model for network intrusion or attack categorization by cleverly combining CNNs (for spatial feature extraction), LSTMs (for sequence modelling), and Attention (for dynamic feature weighting). The model concentrates on the most instructive aspects of the data during categorization, and the multi-pathway design guarantees that both short-term and long-term relationships are recorded.

B. OPTIMIZATION OF HYBRID CNN-LSTM WITH ATTENTION MECHANISM

The CNN, LSTM, and attention layers are the three main components of the suggested SDN threat identification model, and their careful architectural design and rigorous optimization are essential to its successful operation. To improve the model's classification accuracy, reduce training loss, and guarantee generalization to unknown traffic patterns, a thorough optimization approach was used. A combination of grid search, Bayesian optimization, and manual trial-and-error methods were used to tune the hyperparameters. To strike a balance between computing efficiency and feature learning, the number of convolutional filters was adjusted, ranging from 32 to 64. In order to efficiently capture local spatial patterns in the network flow, a kernel size of three was used. In order for the model to keep longer temporal dependencies—which are essential for identifying slow and growing threats like infiltration or low-rate DDoS attacks—the number of units for the LSTM layers varied from 64 to 128. To counteract overfitting, dropout levels between 0.2 and 0.5 were evaluated and applied to both CNN and LSTM layers. The LSTM outputs and the attention mechanism were designed to cooperate. It created context vectors that dynamically highlighted the most significant time steps in the input sequence using trainable weight matrices. In addition to enhancing classification performance, this made the model's decision-making process more interpretable and made it easier to determine which attributes were most important for attack detection. The Adam optimizer, which showed lower validation loss and faster convergence than stochastic gradient descent (SGD), was used for training with a learning rate of 0.0005. The model was trained across 50 to 100 epochs with an early stopping technique to prevent overtraining, and a batch size of 256 was utilized. The Synthetic Minority Oversampling Technique (SMOTE) was applied to the CICIDS 2017 dataset in order to rectify class imbalance, a prevalent problem in intrusion detection datasets. This prevented bias toward the majority

(benign) class and guaranteed that minority classes, such infiltration, botnet, or brute force attacks, were adequately represented. This allowed the model to learn from all categories equally. All input characteristics were also subjected to Min-Max normalization in order to stabilize gradient updates and accelerate convergence.



Categorical class labels were converted into integer representations appropriate for SoftMax classification by the use of label encoding. The optimized hybrid CNN-LSTM-Attention model outperformed both solo deep learning models and conventional machine learning techniques, achieving a stable accuracy of 93.43% despite the attack data's complexity and class unpredictability. This degree of accuracy shows that the model can accurately, scalable, and reliably detect a variety of SDN attack types, which makes it appropriate for real-time intrusion detection deployments in next-generation network infrastructures.

FIGURE 2. Model Configuration VS Performance Metrics

Comparison of five distinct model configurations, each assessed according to four crucial criteria: test accuracy, validation accuracy, batch size, and training epochs. The number of epochs in the first subplot differs greatly between configurations, with Configuration 3 using the most (100 epochs) and Configurations 1 and 5 using the fewest (20 epochs). This variation shows how training time affects model performance. The batch size utilized for training is depicted in the second subplot. Different computational techniques are highlighted by the fact that Configuration 1 utilizes the smallest batch size (512), while Configuration 4 uses the greatest (2048). Validation accuracy is shown in the third subplot, with Configuration 4 achieving the highest value (~93%), suggesting better generalization during model tweaking.

The test accuracy results are finally shown in the fourth subplot, where Configuration 4 once more outperforms the others (~93.2%), indicating good model generalization on unknown data. It's interesting to note that, even with the longest training period, Configuration 3 does not produce the best results, suggesting that using too many epochs does not always improve accuracy and can instead result in overfitting or inefficient training.

A comparative efficiency analysis was carried out against a number of baseline models, including standard CNN, LSTM, CNN-LSTM without attention, Random Forest, and Support Vector Machine (SVM), in order to verify the effectiveness of the suggested hybrid CNN-LSTM model combined with an attention mechanism. The suggested model outperforms all other methods with the maximum classification accuracy of 93.43%, according to the data. Additionally, it performs better in terms of precision (92.60%), recall (92.20%), and F1-score (92.40%), all of which are essential for preserving high detection rates in SDN contexts while reducing false positives.

Traditional CNNs have an average accuracy of 87.65% and are excellent at extracting spatial information, but they are unable to identify temporal connections in sequential network data. With a somewhat higher accuracy of 88.40%, LSTM models are unable to identify spatial flow patterns, despite their ability to represent time-based behaviour. By utilizing both spatial and temporal information, the CNN-LSTM combination without attention increased accuracy to 91.25%; nevertheless, it was still unable to selectively focus on significant features. The suggested model improves detection precision and interpretability by dynamically emphasizing pertinent time steps in the input sequence through the integration of an attention mechanism. The suggested deep learning architecture exhibits notable gains over conventional machine learning models that have trouble handling complicated and unbalanced traffic patterns, such as Random Forest (84.10%) and SVM (85.30%). Furthermore, the model retains a manageable training period (~72 seconds) in spite of its architectural complexity, which qualifies it for real-time or near-real-time

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	Training Time (s)
Traditional CNN	87.65	85.32	86.50	85.90	45

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	Training Time (s)
Traditional LSTM	88.40	86.12	87.05	86.58	60
CNN + LSTM (No Attention)	91.25	89.80	90.50	90.14	68
CNN + LSTM + Attention (Proposed)	93.43	92.60	92.20	92.40	72
Random Forest Classifier	84.10	82.00	83.50	82.74	40
SVM (RBF Kernel)	85.30	83.40	84.70	84.04	95

Figure 3. Efficiency Table

deployment in SDN controllers. For identifying various and changing SDN-based cyberattacks, the hybrid CNN-LSTM with attention mechanism offers the best overall balance of accuracy, efficiency, and robustness.

4. RESULTS AND DISCUSSIONS

The CICIDS 2017 dataset, which offers a thorough and realistic benchmark encompassing a variety of attack types, including DoS, DDoS, infiltration, brute-force, botnet, and web-based threats, was used to thoroughly assess the suggested SDN attack detection model. To guarantee ideal input circumstances and lessen bias toward dominant classes, the dataset was pre - processed using normalization, label encoding, and SMOTE-based class balancing.

The hybrid CNN-LSTM model with attention mechanism outperformed baseline models like traditional CNN, standalone LSTM, and classical machine learning classifiers like SVM and Random Forest, achieving an overall accuracy of 93.43% during training and testing. High precision (92.60%), recall (92.20%), and F1-score (92.40%) were reported by the model in addition to accuracy, demonstrating its resilience in accurately detecting both common and uncommon attack types with few false positives and false negatives. The attention mechanism's incorporation was essential to improving the model's functionality. It improved the detection of sluggish or covert attacks, which frequently elude conventional detection techniques, by allowing the network to concentrate on the most instructive portions of sequential traffic data. Additionally, the CNN layers collected spatial patterns like burst flow characteristics and packet distribution anomalies, while the LSTM layers recorded temporal behaviour like flow intervals and session sequences. This resulted in a highly expressive representation of SDN traffic. The model's generalizability and adaptability to many threat vectors were demonstrated throughout testing, as it remained consistent across multiple assault categories. The suggested deep learning model continuously offered better classification and learning capabilities than traditional machine learning techniques like Random Forest and SVM, which had trouble scaling with complicated traffic and had lower detection rates in minority classes.

Overall, the experimental findings support the efficacy of integrating attention-based, temporal, and spatial learning for SDN intrusion detection. The model's promise for real-time implementation in SDN controllers, providing improved security, automation, and resilience in contemporary programmable networks, is confirmed by the realized performance. Future additions might include real-time zero-day attack detection using online learning methods and deployment optimization for edge computing utilizing TensorFlow Lite.

5. CONCLUSIONS

A powerful and reliable method for detecting a wide range of network threats relevant to Software-Defined Networking (SDN) systems is the CNN-LSTM with Attention model. Its advanced deep learning architecture, which combines the advantages of convolutional neural networks, long short-term memory networks, and an attention mechanism in a synergistic manner, is its fundamental strength. The model can effectively extract features from unprocessed network flow data thanks to its design, comprehend intricate temporal assault patterns, and dynamically concentrate on the most important signs of hostile behaviour. As a result, network traffic is deeply and contextually understood, surpassing static matching of signatures. The model can identify a wide range of contemporary cyberthreats because it was trained on the extensive and real-world CICIDS2017 dataset, which contains 14 different attack types in addition to benign traffic. 17 Its strong resistance to both frequent and uncommon attack types is further supported by its high accuracy (up to 93.43%) on this unbalanced dataset. The model supports a thorough and integrated security posture for SDN by identifying attacks across several types (volume-based, stealthy, credential-based, and web-based) and phases of the kill chain (reconnaissance, exploitation, and persistence). It is a flexible detection system rather than just a point answer for one kind of attack.

REFERENCES

[1] H. Xue and B. Jing, "SDN Attack Identification Model Based on CNN Algorithm," in IEEE Access, vol. 11, pp. 87652-87666, 2023

[2] E.W. E.Viklund, I. Nilsson, and A. K. Forsman, ‘‘Nordic population-based study on internet use and perceived meaningfulness in later life: How they are linked and why it matters,’’ *Scand. J. Public Health*, vol. 50, no. 3, pp. 381–388, May 2022.

[3] N. Ravi and S. M. Shalinie, ''Black Nurse-SC: A novel attack on SDN controller,'' *IEEE Commun. Lett.*, vol. 25, no. 7, pp. 2146–2150, Jul. 2021.
[4] H. Li, J. Lu, J.Wang, H. Zhao, J. Xu, and X. Chen, ''*SDM4IIoT* : An SDNbased multicast algorithm for industrial Internet of Things,'' *IEICE Trans. Commun.*, vol. 105, no. 5, pp. 545–556, May 2022.
[5] D. Tang, Y. Yan, S. Zhang, J. Chen, and Z. Qin, ''Performance and features: Mitigating the low-rate TCP-targeted DoS attack via SDN,'' *IEEE J. Sel. Areas Commun.*, vol. 40, no. 1, pp. 428–444, Jan. 2022.
[6] S. Ravikumar and D. Kavitha, ''CNN-OHGS: CNN-oppositional-based Henry gas solubility optimization model for autonomous vehicle control system,'' *J. Field Robot.*, vol. 38, no. 7, pp. 967–979, May 2021.
[7] M. A. Ouamri, M. Azni, D. Singh, W. Almughalles, and M. S. A. Muthanna, ''Request delay and survivability optimization for software defined-wide area networking (SD-WAN) using multi-agent deep reinforcement learning,'' *Trans. Emerg. Telecommun. Technol.*, vol. 34, no. 7, Jul. 2023, Art. no. e4776.
[8] M. A. Ouamri, G. Barb, D. Singh, and F. Alexa, ''Load balancing optimization in software-defined wide area networking (SD-WAN) using deep reinforcement learning,'' in *Proc. Int. Symp. Electron. Telecommun. (ISETC)*, Timişoara, Romania, Nov. 2022, pp. 1–6.
[9] S. Garg, K. Kaur, G. Kaddoum, S. H. Ahmed, and D. N. K. Jayakody, ''SDN-based secure and privacy-preserving scheme for vehicular networks: A 5G perspective,'' *IEEE Trans. Veh. Technol.*, vol. 68, no. 9, pp. 8421–8434, Sep. 2019.
[10] S. Badotra and S. N. Panda, ''SNORT based early DDoS detection system using open daylight and open networking operating system in software defined networking,'' *Cluster Compute.*, vol. 24, no. 1, pp. 501–513, Mar. 2021.
[11] U. Ahmed, J. C.-W. Lin, and G. Srivastava, ''A resource allocation deep active learning based on load balancer for network intrusion detection in SDN sensors,'' *Compute. Commun.*, vol. 184, pp. 56–63, Feb. 2022.
[12] A. El Kamel, H. Eltaief, and H. Youssef, ''On-the-fly (D)DoS attack mitigation in SDN using deep neural network-based rate limiting,'' *Compute. Commun.*, vol. 182, pp. 153–169, Jan. 2022.
[13] J. Chen, L. Wang, and S. Duan, ''A mixed-kernel, variable-dimension memristive CNN for electronic nose recognition,'' *Neurocomputing*, vol. 461, pp. 129–136, Oct. 2021.
[14] R. Bao and Z. Yang, ''CNN-based regional people counting algorithm exploiting multi-scale range-time maps with an IR-UWB radar,'' *IEEE Sensors J.*, vol. 21, no. 12, pp. 13704–13713, Jun. 2021.
[15] S. Chen, H. Pei, J. Pisonero, S. Yang, Q. Fan, X. Wang, and Y. Duan, ''Simultaneous determination of lithology and major elements in rocks using laser-induced breakdown spectroscopy (LIBS) coupled with a deep convolutional neural network,'' *J. Anal. At. Spectrometry*, vol. 37, no. 3, pp. 508–516, 2022.
[16] S. Guo, Z. Wang, Y. Lou, X. Li, and H. Lin, ''Detection method of photovoltaic panel defect based on improved mask R-CNN,'' *J. Internet Technol.*, vol. 23, no. 2, pp. 397–406, Mar. 2022.
[17] Y. Xue, Y. Wang, J. Liang, and A. Slowik, ''A self-adaptive mutation neural architecture search algorithm based on blocks,'' *IEEE Compute. Intell. Mag.*, vol. 16, no. 3, pp. 67–78, Aug. 2021.
[18] S. Siddiqui, S. Hameed, S. A. Shah, I. Ahmad, A. Aneiba, D. Draheim, and S. Dustdar, ''Toward software-defined networking-based IoT frameworks: A systematic literature review, taxonomy, open challenges and prospects,'' *IEEE Access*, vol. 10, pp. 70850–70901, 2022.
[19] M. Khalid, S. Hameed, A. Qadir, S. A. Shah, and D. Draheim, ''Towards SDN-based smart contract solution for IoT access control,'' *Compute. Commun.*, vol. 198, pp. 1–31, Jan. 2023.
[20] K. Renuka, D. S. Roy, and K. H. K. Reddy, ''An SDN empowered location aware routing for energy efficient next generation vehicular networks,'' *IET Intell. Transp. Syst.*, vol. 15, no. 2, pp. 308–319, Feb. 2021.