iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 **Issue:** VIII **Month of publication:** August 2025

DOI: https://doi.org/10.22214/ijraset.2025.73891

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue VIII Aug 2025- Available at www.ijraset.com

A Survey on Proactive Security in Software Defined Networks Using a Convolutional Neural Network-Based Early Warning System

Shwetha A B[1], Likhitha V[2], Hithaishi G[3], Bindushree E[4], Nanditha R[5]
[1]*Assistant Professor, Dept of CSE, Sapthagiri College of Engineering*
[2, 3, 4, 5]*Dept of CSE, Sapthagiri College of Engineering*

Abstract: ***Software-Defined Networking (SDN) centralizes control and improves programmability, but this also increases vulnerability to large-scale threats such as DDoS. Conventional intrusion detection systems often fail in this context due to static rules and high false alarms. Recent studies apply deep learning for anomaly detection, with Convolutional Neural Networks (CNNs) offering faster training and lower latency than recurrent or hybrid models. This survey reviews CNN-based IDS approaches and compares them with alternatives like DNNs [1], GRU-RNNs [2], and SAE-based models [3]. Our analysis highlights CNNs' faster convergence and adaptability for real-time detection, while also identifying the challenges of scalability, data diversity, and deployment overhead. The paper concludes by outlining open research directions toward intelligent, lightweight, and automated IDS solutions for strengthening SDN environments.***

I. INTRODUCTION

Modern networks face increasing complexity and scalability issues that traditional methods struggle to handle. SDN addresses this by introducing centralized, programmable control through the separation of control and data planes. This architectural shift simplifies configuration, enables rapid service deployment, and provides global traffic visibility. At the same time, however, the reliance on a centralized controller exposes SDN to new forms of security risks, with the control plane becoming a prime target for cyberattacks.

Among the most critical threats are Distributed Denial of Service (DDoS) attacks and other flow-based intrusions capable of overwhelming the controller and destabilizing the network. Unlike traditional distributed networks, the single point of control in SDN makes such attacks more impactful, underscoring the need for proactive and resilient defense strategies.

Intrusion Detection Systems (IDS) remain a core element of network protection, designed to identify abnormal or malicious behavior. Signature-based IDS can recognize attacks only if the patterns are already in their database, making them ineffective against zero-day threats. In contrast, anomaly-based IDS detect unusual traffic behaviors, enabling them to spot previously unknown attacks. Yet, they often face drawbacks such as high false-positive rates and heavy reliance on handcrafted features, which reduce adaptability in dynamic SDN environments.

Machine learning (ML) and deep learning (DL) methods are increasingly applied to IDS in SDNs as they can model traffic behavior and boost detection accuracy. Prior studies have demonstrated that deep learning models can effectively capture traffic patterns and improve detection accuracy.

As an example, [1,2] explored the use of DNNs alongside GRU-based models for SDN-based intrusion detection, achieving high detection rates on benchmark datasets. [3] introduced a Stacked Autoencoder (SAE) for DDoS detection, while [4] combined neural models with Snort to enhance real-time responses. More recently, [5] proposed a hybrid CNN–LSTM approach that integrates spatial and temporal feature learning for anomaly detection, improving accuracy across multiple attack types. Although effective, recurrent and hybrid models often demand high training time and computational resources, which limits their suitability for real-time SDN environments.

This survey highlights how CNNs—initially created for image classification—have been adapted for SDN traffic monitoring. CNNs are attractive in this domain because they extract spatial patterns from structured data, provide faster training and inference in contrast to alternative deep learning methods. By reviewing CNN-based intrusion detection in SDNs and comparing them with other deep learning approaches, this paper aims to present their strengths, limitations, and the future hurdles that must be overcome to achieve lightweight, scalable, and intelligent IDS solutions for SDN environments.

©IJRASET: All Rights are Reserved

II. BACKGROUND

A. Problem Description

The increasing adoption of Software-Defined Networking (SDN) provides clear advantages such as centralized control, programmability, and easier management. At the same time, this centralization introduces security risks, as attackers can exploit the controller to launch various threats including flooding attacks, flow saturation, or even controller hijacking.

Conventional Intrusion Detection Systems (IDS) struggle to address these challenges. Signature-based IDS methods are limited to detecting only previously known attack patterns, leaving networks exposed to zero-day exploits. On the other hand, anomaly-based IDS approaches can recognize unfamiliar behaviors by identifying deviations from normal traffic. However, they are prone to high false alarm rates and often depend heavily on manual feature engineering, reducing their adaptability in highly dynamic SDN environments.

To mitigate these issues, recent work leverages ML and DL frameworks. Models such as DNNs, RNNs, and GRUs demonstrate clear advantages compared with legacy intrusion detection strategies. While performance has improved, deep models typically consume significant resources and involve prolonged training phases, making real-time deployment in live SDN infrastructures challenging.

As a result, there is growing interest in building lightweight, scalable, and efficient intrusion detection systems capable of detecting both known and emerging threats in SDN environments. The use of CNNs in anomaly detection has gained traction, as they can identify spatial dependencies within structured flows while operating with reduced delay and complexity.

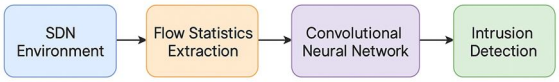


Fig1.System Overview

B. Adversary Model

In this context, adversaries are considered adaptive and intelligent actors who attempt to degrade network performance or compromise availability by exploiting SDN's centralized architecture. Common strategies include packet flooding attacks, where multiple compromised devices overwhelm the controller or switches, and flow table exhaustion, where attackers generate many unique flows to deplete the limited table capacity of SDN switches. More sophisticated adversaries may launch stealthy or low-rate intrusions, imitating legitimate traffic to bypass detection mechanisms that rely on thresholds or simple statistical models. Attackers may also employ zero-day strategies, using previously unseen attack techniques that evade signature-based defenses. These attacks are especially dangerous as they may persist unnoticed over extended durations, enabling sustained control over network resources.

An effective IDS for SDN must therefore be capable of detecting diverse attacks in near real-time, learning complex traffic patterns as they evolve, and responding proactively to prevent widespread disruption.

III. LITERATURE SURVEY

A. A deep learning-based IDS for SDN was introduced by Tang et al. [1]

Tang, T. A. et al. (2018)

Employed a DNN architecture, with training performed on the NSL-KDD dataset using only six statistical flow-based features. The model demonstrated that deep models could learn complex traffic behavior and identify effective intrusions with reduced preprocessing.

Advantage: Achieves high accuracy using a minimal feature set; demonstrates potential of DL in SDN environments.
Limitation: High training time and limited real-time deployment capability due to dense architecture and overfitting risks.

B. Deep Recurrent Neural Network for Intrusion Detection in SDN-Based Networks was introduced by Tang, T. A. et al. [2]

Tang, T. A. et al. (2018)

Introduced a GRU-based RNN architecture to model temporal flow dependencies in network traffic. The approach-maintained state information to detect evolving and sequential attack patterns.

Advantage: Effectively models time-based intrusions like slow-rate DDoS; improved accuracy (~89%) over static models.
Limitation: Inference speed is limited due to sequential computation; unsuitable for real-time applications in high-speed SDNs.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

C. A Deep Learning-Based DDoS Detection System in Software-Defined Networking was introduced by Nguyen, T. A. & Kim, G. [2]

Nguyen, T. A. & Kim, G. (2018)

Developed a Stacked Autoencoder (SAE)-based model to detect multi-vector DDoS attacks in SDNs. The system learns unsupervised traffic patterns and classifies based on reconstruction errors.

Advantage: Automatically learns non-linear relationships without extensive feature engineering; high detection rate, Unsupervised learning capability enables detection of novel and zero-day attack patterns.

Limitation: Complex training process and high computational overhead; lacks responsiveness to real-time threats.

D. Machine Learning Based Intrusion Detection System for Software Defined Networks was introduced by Abubakar, A. & Pranggono, B [4]

Abubakar, A. & Pranggono, B. (2017)

Proposed a hybrid IDS combining Snort with a neural network for enhanced detection. Implemented in a virtual SDN environment for practical evaluation.

Advantage: Integrates traditional signature-based IDS with anomaly detection; improves detection of unknown threats, supports real-time packet analysis through Snort integration, facilitating early threat identification.

Limitation: Still partially rule-dependent; limited adaptability to zero-day attacks and lacks temporal analysis.

E. Flow-Based Anomaly Detection in Software Defined Networks Using Hybrid Deep Learning Model was introduced by Nguyen, T. A. et al. [5]

Nguyen, T. A. et al. (2020)

Presented a hybrid architecture combining GRU and LSTM layers, enhanced with ANOVA F-test and Recursive Feature Elimination (RFE) for optimized feature selection. This design aimed to better capture both short- and long-term dependencies in traffic.

Advantage: Captures both short-term and long-term flow dependencies with improved accuracy; dimensionality reduction enhances efficiency.

Limitation: Dual recurrent structure increases complexity and latency; sequential processing limits real-time deployment feasibility.

IV. METHODOLOGY

A. Data Sources and Traffic Representation

The survey indicates that benchmark datasets are widely used, with NSL-KDD dominating due to its well-balanced distribution of attack and normal traffic instances across categories such as DoS, probe, R2L, and U2R. Researchers often extract flow-level attributes that align with the information available to SDN controllers, including features like packet counts, flow duration, and byte statistics. Some studies utilized the entire feature set of 41 attributes, while others selected a smaller subset—typically six to twelve variables—based on relevance, reducing overhead and enabling models to operate closer to real-time conditions. Beyond static datasets, several papers employed Mininet-based SDN testbeds to generate synthetic attack traces (e.g., TCP, UDP, or ICMP floods).

B. Preprocessing and Feature Engineering

Before training, input data was systematically prepared to improve model performance. Continuous attributes were normalized using techniques such as min–max scaling or z-score standardization to maintain consistency across features. Encoding methods were applied to categorical inputs, including protocol types and flag indicators, to represent them numerically like label encoding or one-hot encoding.

To further refine the data, some works addressed class imbalance through oversampling or undersampling strategies, while others applied dimensionality reduction methods. A number of hybrid approaches used ANOVA F-test and RFE to select the most important features, thereby lowering complexity and reducing training time.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

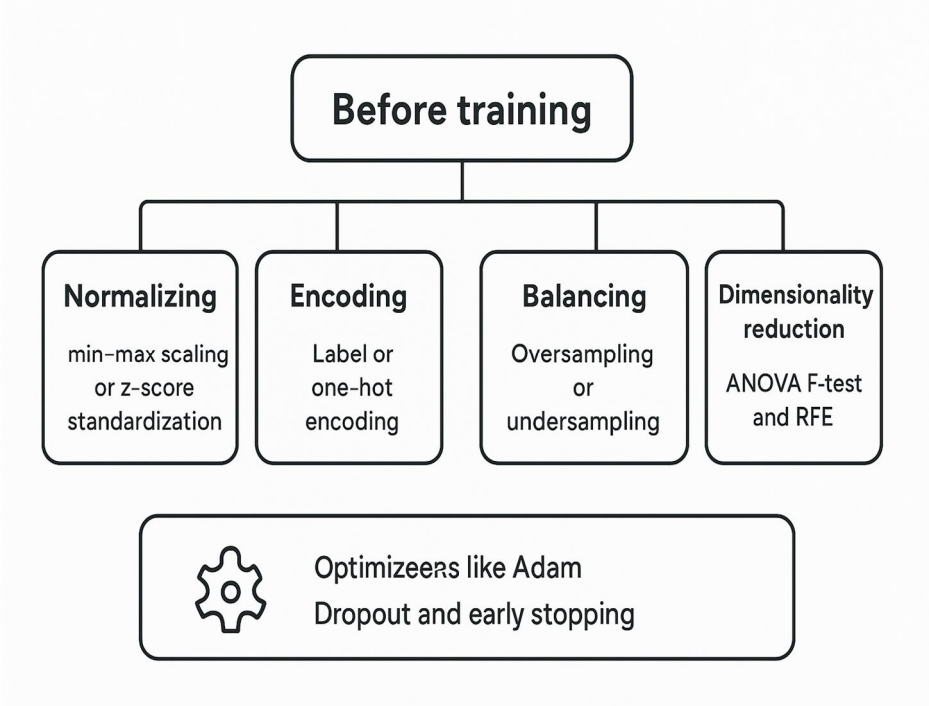


Fig 2. Pipeline for Flow-Based Intrusion Detection.

C. Model Architectures and Learning Strategies

Surveyed studies explored different DL models, such as DNNs built with several dense layers to capture hierarchical traffic patterns. Recurrent Neural Networks (RNNs), particularly GRU-based variants, were applied to learn temporal dependencies within sequential flows, making them effective against slow or time-dependent intrusions. Stacked Autoencoders (SAEs) were utilized in unsupervised scenarios, relying on reconstruction errors to identify abnormal traffic. More advanced works combined these methods into hybrid frameworks, such as GRU–LSTM architectures or IDS models that integrated deep learning with rule-based systems like Snort, thereby blending anomaly detection with signature verification. Training strategies frequently employed optimizers like Adam, along with techniques such as dropout and early stopping, to mitigate overfitting.

D. Evaluation Frameworks and Metrics

To evaluate performance, accuracy was taken as the overall benchmark, while precision, recall, and F1-score were used to give a clearer picture of how well the system handled different types of traffic under class imbalance and to prevent bias toward dominant attack categories. Confusion matrices were commonly reported to visualize classification outcomes across multiple attack classes, showing distributions of true and false positives and negatives. In addition to accuracy-related metrics, computational aspects such as training duration, inference latency, and memory footprint were analyzed to determine the feasibility of real-time deployment. Some experiments also evaluated IDS responsiveness by embedding models into SDN controllers (e.g., POX, Ryu) and measuring decision times on live flow data.

E. *Deployment Considerations and Limitations*

Although many deep learning approaches achieved strong results on benchmark datasets, practical deployment revealed several constraints. Recurrent models like GRUs and LSTMs faced bottlenecks due to sequential computation, limiting throughput in high-speed SDN environments. SAE-based solutions, though effective against multi-vector attacks, demanded intensive pretraining and tuning. Hybrid frameworks introduced additional layers of complexity, making real-time integration challenging. Moreover, models trained on static datasets often lacked adaptability to evolving threats such as zero-day or polymorphic attacks. These shortcomings point to the need for lightweight, parallelizable, and deployment-friendly architectures, leading to growing attention on CNNs as promising solutions for instantaneous anomaly detection in SDNs, owing to their faster inference and suitability for large-scale traffic monitoring.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

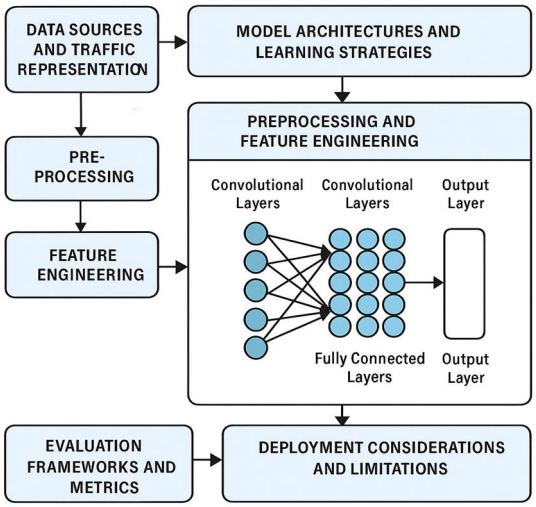


Fig 3. Pipeline for Flow-Based Intrusion Detection.

V. FUTURE WORKS

1) Real-time SDN traffic integration with online learning
2) Lightweight DL models for edge/IoT deployment
3) Unified frameworks for multi-class and multi-stage attack detection
4) Incorporation of temporal–spatial traffic features
5) Cross-dataset validation and transfer learning approaches
6) Explainable and interpretable IDS models

VI. CONCLUSION

Securing Software Defined Networks (SDNs) against flow-based intrusions has become increasingly important due to their centralized and programmable nature. Deep learning approaches, including DNNs, GRUs, SAEs, and hybrid models, have shown promise in detecting a wide range of threats by learning complex traffic patterns. However, these models often face challenges in terms of computational complexity, scalability, and real-time responsiveness. Recent interest in Convolutional Neural Networks (CNNs) reflects a shift toward more efficient, parallelizable architecture suitable for real-time deployment. While promising, further work is needed to enhance model robustness, interpretability, and adaptability to evolving attack patterns. Future efforts should also focus on improving dataset quality, reducing overhead, and ensuring seamless integration within SDN environment.

REFERENCES

[1] T. A. Tang, L. Mhamdi, D. McLernon, S. A. R. Zaidi, and M. Ghogho, “Deep learning approach for Network Intrusion Detection in Software Defined Networking,” in Proc. Int. Conf. Wireless Networks and Mobile Communications (WINCOM), 2016, pp. 258–263, doi: 10.1109/WINCOM.2016.7777224.
[2] T. A. Tang, L. Mhamdi, D. McLernon, S. A. R. Zaidi, and M. Ghogho, “Deep Recurrent Neural Network for Intrusion Detection in SDN-based Networks,” in Proc. IEEE Int. Conf. Advanced Networks and Telecomm. Systems (ANTS), 2018, pp. 1–6, doi: 10.1109/ANTS.2018.8710098.
[3] T. A. Nguyen and G. Kim, “A Deep Learning-Based DDoS Detection System in Software-Defined Networking,” in Proc. 2018 IEEE Int. Conf. Information Networking (ICOIN), 2018, pp. 1–5, doi: 10.1109/ICOIN.2018.8343163.
[4] A. Abubakar and B. Pranggono, “Machine Learning Based Intrusion Detection System for Software Defined Networks,” in Proc. 2017 Seventh Int. Conf. Emerging Security Technologies (EST), 2017, pp. 138–143, doi: 10.1109/EST.2017.8090413.
[5] T. A. Nguyen, N. D. Nguyen, and D. Tran, “Flow-Based Anomaly Detection in Software-Defined Networking Using Hybrid Deep Learning Model,” IEEE Access, vol. 8, pp. 32536–32545, 2020, doi: 10.1109/ACCESS.2020.2973560.
[6] J. Kim, H. Kim, M. Shim, and E. Choi, ‘‘CNN-based network intrusion detection against denial-of-service attacks,’ Electronics, vol. 9, p. 916, Jun. 2020.
[7] R. Palanikumar and K. Ramasamy, ‘‘Software defined network based self diagnosing faulty node detection scheme for surveillance applications,’’ Compute. Commun., vol. 152, pp. 333–337, Feb. 2020.
[8] Y. Goto, B. Ng, W. K. G. Seah, and Y. Takahashi, ‘‘Queueing analysis of software defined network with realistic OpenFlow–based switch model,’’ Compute. Netw., vol. 164, Dec. 2019, Art. no. 106892.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

[9] A. Shaghaghi, M. A. Kaafar, R. Buyya, and S. Jha, ‘‘Software-defined network (SDN) data plane security: Issues, solutions, and future directions,’’ in Handbook of Computer Networks and Cyber Security, B. Gupta, G. Perez, D. Agrawal, and D. Gupta, Eds. Cham, Switzerland: Springer, 2020, doi: 10.1007/978-3-030-22277-2_14.

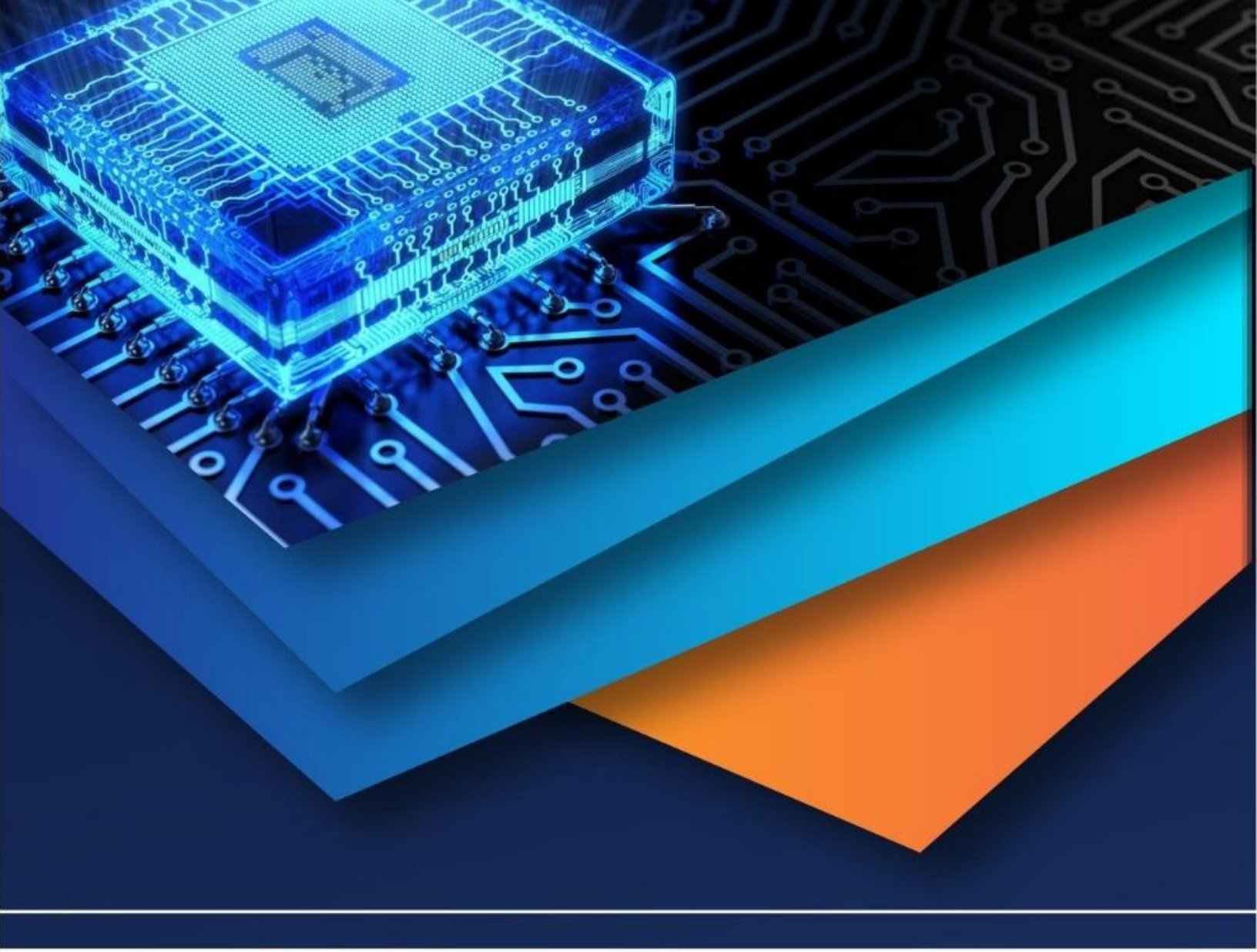
[10] K. Bhushan and B. B. Gupta, ‘‘Distributed denial of service (DDoS) attack mitigation in software defined network (SDN)-based cloud computing environment,’’ J. Ambient Intell. Hum. Comput., vol. 10, no. 5, pp. 1985–1997, May 2019.

[11] I. Ahmad, S. Namal, M. Ylianttila, and A. Gurtov, ‘‘Security in software defined networks: A survey,’’ IEEE Commun. Surveys Tuts., vol. 17, no. 4, pp. 2317–2346, 4th Quart., 2015.

[12] H. Wang and W. Li, ‘‘DDosTC: A transformer-based network attack detection hybrid mechanism in SDN,’’ Sensors, vol. 21, no. 15, p. 5047, Jul. 2021.

[13] S. Boukria and M. Guerroumi, ‘‘Intrusion detection system for SDN network using deep learning approach,’’ in Proc. Int. Conf. Theor. Applicative Aspects Comput. Sci. (ICTAACS), Dec. 2019, pp. 1–6.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)