International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 10 Issue: 04 | April - 2026 SJIF Rating: 8.659 ISSN: 2582-3930

Experiments with Digital Security Processes over SDN-based Cloud-native 5G Core Networks

1st Golla Shivani
Department of Electronics and Communication Engineering
Institute of Aeronautical Engineering
Hyderabad, India
shivanigolla030@gmail.com

2nd Ms. Y. Meghamala
Department of Electronics and Communication Engineering
Institute of Aeronautical Engineering
Hyderabad, India
y.meghamala@iare.ac.in

3rd Jakkidi Sai Tejaswini
Department of Electronics and Communication Engineering
Institute of Aeronautical Engineering
Hyderabad, India
22951a04f7@iare.ac.in

4th D. Rohan Abhiram
Department of Electronics and Communication Engineering
Institute of Aeronautical Engineering
Hyderabad, India
22951a04e3@iare.ac.in

***Abstract*—An SDN-based 5G ecosystem to improve 5G network security is presented in this paper. The following fundamental containerized 5G network functions are integrated with SDNs by the system: (i) monitoring modules that take into account network statistics and compute resources; (ii) attack detection mechanisms based on both basic statistical and artificial intelligence models; and (iii) security events modules that facilitate the establishment of access control rules. For the SDN system, DoS attack development, detection, and avoidance experiments are presented.**

***Index Terms*—5G core networks, software-defined networking, cloud-native architecture, AI-driven intrusion detection, DDoS mitigation, network traffic analytics, deep learning models, SDN-based monitoring, intelligent security frameworks, and cyber-physical resilience.**

I. Introduction

The development of 5G networks is a major step toward enabling large-scale digital services that need extremely low latency, high bandwidth, and continuous connectivity. These features are made possible by the shift from traditional hardware-based architectures to cloud-native and software-controlled network environments. Because network functions that were previously fixed in physical equipment are now provided as virtualized services and microservices, the system is more flexible, scalable, and manageable. However, this change also exposes the network to new kinds of security vulnerabilities because software-based components and programmable interfaces are more vulnerable to abuse and targeted attacks. Large-scale digital applications that need constant connectivity, high capacity, and extremely low latency will be made possible by the development of 5G networks. These opportunities are made possible by the replacement of conventional hardware-based architectures by cloud-native and software-controlled network infrastructures. The system is now more scalable, flexible, and manageable thanks to the implementation of network functions that were previously fixed in physical hardware as microservices and virtualized services. However,

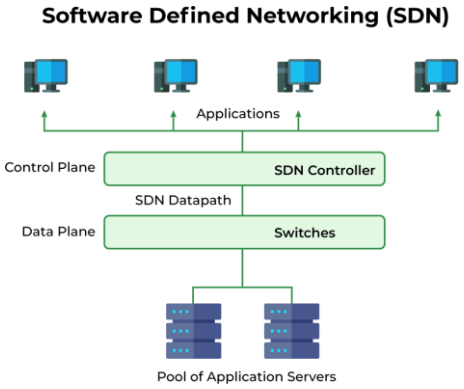

Fig. 1. Software-defined networking architecture.

this shift also exposes the network to new kinds of security flaws because software-based components and programmable interfaces are more vulnerable to misuse and targeted attacks. To address these problems, this work proposes a hybrid security system that combines SDN programmability with AI-based and data-driven threat detection techniques. Instead of relying solely on one method, the system evaluates network traffic patterns using both deep learning architectures and lightweight statistical models. A publicly available SDN–5G dataset is used to replicate realistic scenarios, enabling the evaluation of models such as EMA, ARIMA, MLP, CNN1D, and CNN2D. The study claims that because CNN2D can recognize complex relationships and patterns present in modern 5G data, it has the highest accuracy. In addition to detecting attacks, the system's decision-making module automatically reacts to them. When malicious activity is identified, the SDN controller takes the appropriate action, such as packet dropping or blocking. This preserves network dependability and stops attacks from propagating among linked components. Because

the entire framework is implemented as a web application, users can upload traffic samples, view detection results, and control security settings, making the system easily accessible and testable. When programmable SDN control is combined with AI-based detection, cloud-native 5G network security is generally greatly improved. Smart and automated security solutions like the one suggested here will be crucial for preserving reliable and robust communication infrastructure as 5G supports essential services and expansive digital ecosystems.

II. Literature Survey (Related work)

The authors of [1] give a thorough summary of the potential security risks associated with 5G and future network generations. According to their research, the shift to software-driven, virtualized systems—especially those that make use of edge computing, SDN, and NFV—increases the attack surface. They argue for intelligent, automated, and context-aware solutions because traditional security measures are inadequate in such dynamic environments. The study also shows how programmable network designs, such as SDN, can significantly lessen the severity of complex attacks by using quick detection and response when paired with AI-based analysis.

A load-balancing system for microservices running in 5G-scale scenarios is demonstrated by the authors in [2]. They point out that the elasticity and spread of microservice deployment are too much for previous load-balancing solutions to handle. The suggested solution improves traffic distribution and slightly boosts resistance against congestion-based attacks and service interruptions by dynamically altering routing paths using SDN concepts.

The paper discussed in [3] provides a comprehensive explanation of how SDN helps to separate the control and data planes, thereby enabling improved network programmability, using a softwarized perspective of 5G design. The authors claim that because network operators can dynamically reconfigure services based on demand or the threat environment, such a design improves scalability and makes security rule implementation easier. Their findings unequivocally indicate that SDN ought to play a major role in cloud-native 5G security systems.

Reference [4] presents a real-world illustration of SDN controller operation using open-source tools accessible on GitHub. Real-life cases highlight how SDN controllers manage flow rules, monitor traffic, and respond to network events by use of actual samples. Although the primary goal of the presentation is educational, it is very helpful for understanding how controllers operate in real-time contexts. Furthermore, it offers guidance on how to build test conditions without specialized equipment, which is quite useful in experimental research like this one.

In [5], the Floodlight SDN Controller is shown as a flexible and often used instrument for research and development of SDN applications. The authors detail Floodlight's customizable packet forwarding, traffic inspection, and intrusion detection logic. Its modular design lets several outside analytics engines—including machine learning models—be integrated, therefore automating threat-mitigation operations is reasonable. Given that programmable rules and artificial intelligence models have to interact, these characteristics make Floodlight appropriate for the goals of the proposed system.

The technique in [6] explores a different path for malware traffic identification by means of deep neural networks for classification and transforming PCAP data into visual representations. By showing network flows as pictures, the authors demonstrate how CNN-based systems can spot hidden structural relationships that might not be clear from conventional numerical characteristics. Their findings support the idea that in complex traffic conditions, 1D and 2D convolutional neural networks can significantly enhance attack detection. This perspective shapes directly the drive to evaluate CNN1D and CNN2D models in the present study.

[7] lastly discusses the Malaga-UMA 6G testbed, which offers a flexible environment for testing next-generation communication technologies. Although the testbed is mostly concentrated on 6G research, it provides crucial insights on how cloud-native architectures, artificial intelligence modules, and SDN controllers could be implemented together in real setups. Large-scale experimental platforms help to demonstrate how automatic testing, monitoring, and evaluation of security systems may be assisted by design that is With regard to practical application and verification, this inspires possible future additions to the proposed system.

III. System Architecture

The proposed experimental environment's two primary functional components are the Network Environment and the Attack Identification Engine (AIE). The AIE, which serves as the centralized control module, is responsible for configuring network functions, collecting monitoring data, identifying anomalous activity, and executing security measures. Attack-generating modules, simulated user devices, and containerized 5G core services comprise the Network Environment, which is used to test the resilience of the system.

A. Cloud-Native 5G Core Network

The fully containerized design of the testbed's 5G core replicates modern cloud-native deployments. Important network functions like AMF, SMF, UPF, and authentication services are instantiated as separate containers using the free5GC framework. The 5G core functions are housed in the remaining nine containers, two of which are used as the User Equipment (UE) and radio access node. By enabling communication between all components, SDN-based virtual switches allow for dynamic routing and centralized packet flow control. Modularity, scalability, and straightforward integration with the monitoring and detection subsystems are guaranteed by this configuration. The management interface of the SDN controller, which shows the logical network architecture and flow-level interactions, enables real-time system visibility.

B. Attack Generation Engine

The Attack Generation Engine (AGE) is responsible for producing controlled harmful traffic in order to evaluate system performance under adversarial conditions. It contains several tools that can launch a range of attacks, including denial-of-service (DoS) attempts, protocol-level floods, vulnerability scanning, and API injection attacks. In addition to traditional attack methods, the engine has neural network-driven generators that can produce a variety of flexible malicious packet streams. As a result, both static and dynamic threat patterns can be compared to the detection models.

C. SDN Controller and Monitoring Framework

The SDN controller is the main hub for making decisions regarding packet forwarding, rule installation, and policy enforcement throughout the network. It incorporates two primary components:

1) Monitoring Broker:: responsible for gathering real-time data on the resource consumption, packet counts, flow duration, and link utilization of every network function. This information supports anomaly detection and traffic profiling.

2) Security Module:: implements flow-control policies using the alarm signals from the detecting mechanisms. Depending on the identified threat, the module can limit bandwidth, stop flows, or impose different mitigation rules across SDN switches. The SDN controller platform is Floodlight; OpenFlow is the southbound protocol for rule management; OpenvSwitch is used for virtual switching. The Docker monitoring interface provides container-level computer measurements.

D. Detection Mechanisms

Two supporting layers built into the detection layer are an AI-based detection module and a statistical detection module.

1) Statistical Detection Module: Among the statistical techniques employed are Exponential Moving Average (EMA) and Auto Regressive Integrated Moving Average (ARIMA), which serve as baseline models for contrast with AI-based methods, and change-point detection, threshold-based detection, and clustering-based analysis to detect anomalous behavior.

2) AI-Based Detection Module: The AI-driven subsystem uses packet-capture (PCAP) data. Two preprocessing methods are employed: • Transformation of unprocessed packet flows into sequences with integer encoding • extraction of flow-based statistical characteristics, including inter-arrival periods, average packet size, and packet rate Two deep learning systems, a 1D Convolutional Neural Network (CNN1D) and a These representations are used to train Multilayer Perceptrons (MLPs). Every model identifies particular traffic patterns indicating different kinds of attacks. During inference, the detection module reviews incoming traffic and produces alerts should it discover suspicious activity.

E. Decision-Making Algorithm

The Decision-Making Algorithm (DA) examines the outputs of the detection module and then decides and puts in place the right mitigating steps. Upon spotting an attack, the DA works

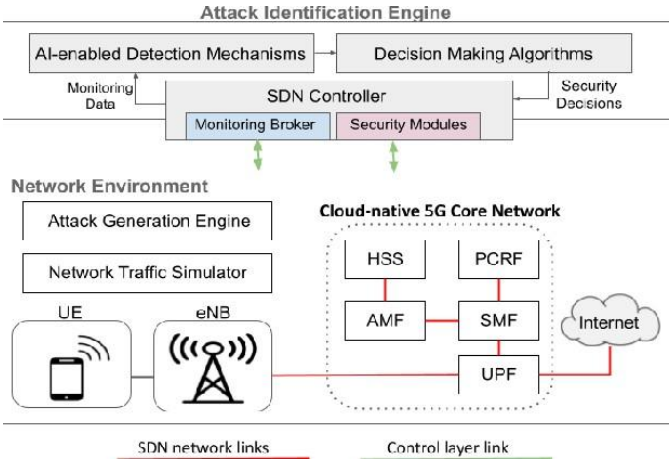

Fig. 2. Overview of the proposed solution

with the SDN controller to apply filtering or blocking rules after gathering essential information, including the IP address of the attacker. These actions will help you to minimize service interruptions and stop hostile traffic from infecting the network.

F. Network Traffic Simulation

Two solutions are employed to replicate network traffic and user behavior for testing purposes:

1) Apache JMeter:: Generates fake traffic patterns to look at performance measures including packet loss, throughput, and latency across a range of attack and stress situations.

2) OpenAirSim:: Simulating user equipment and radio access activities helps to produce realistic traffic and mobility patterns that rather resemble real 5G installations. These technologies enable the performance of controlled studies evaluating system behavior as well as the consequences of several assault situations.

IV. Proposed Methodology

By combining Software-Defined Networking (SDN) with statistical and AI-based attack detection techniques, the suggested methodology seeks to enhance the security of cloud-native 5G networks. To reduce risks like Distributed Denial-of-Service (DDoS) attacks, the strategy heavily emphasizes intelligent attack detection, automated decision-making, and continuous traffic monitoring. Data collection, preprocessing, traffic monitoring, attack detection using various models, and decision-based mitigation are among the many stages of the methodology. Below is a detailed description of each step.

A. Dataset Acquisition

Since real-time cloud servers and SDN equipment are not easily accessible, a publicly available SDN–5G network traffic dataset is used for experimentation. The dataset was collected from the Kaggle repository and contains labeled traffic records that represent both typical and DDoS attack scenarios. It includes important network attributes such as flow time, packet length, port number, protocol type, and traffic statistics. This

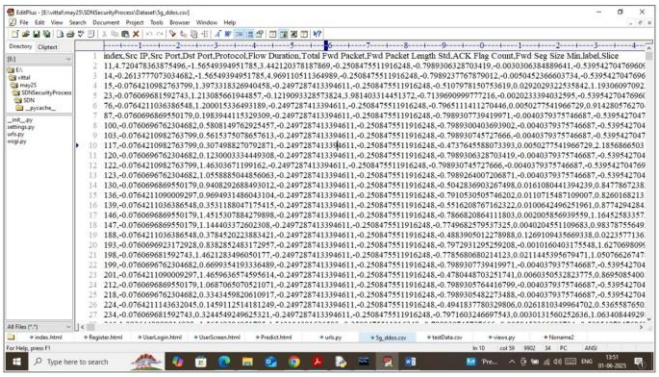

Fig. 3. dataset contains all network information such as protocol type, port no, packet length etc

dataset can be used to train and test the detection models, enabling accurate network behavior simulation.

B. Data Preprocessing

To enhance learning efficacy and data quality, the dataset is preprocessed before the model is trained. This step entails encoding categorical features into numerical form, handling missing data, and normalizing numerical attributes. After processing, 80% of the dataset is used to train the models, and the remaining 20% is reserved for testing and assessment.

C. SDN-Based Traffic Monitoring

The main method for tracking traffic is SDN. Network flow statistics are gathered by the SDN controller, which then sends the monitoring information to the detection engine. The detection system can efficiently evaluate traffic patterns thanks to its centralized monitoring technique, which offers a world wide perspective of network activity. AI-based and statistical detection models both use the monitored data as input.

D. Attack Detection Using Statistical and AI Models

SDN is the main tool used to monitor traffic. Network flow statistics are gathered by the SDN controller, which provides the monitoring data to the detection engine. The detection system can efficiently evaluate traffic patterns thanks to this centralized monitoring approach, which offers a world wide perspective of network activity.The monitored data is the input for both AI-based and statistical detection algorithms.

1) Statistical Models: AutoRegressive Integrated Moving Average (ARIMA) and Exponential Moving Average (EMA) are two examples of baseline statistical models. These models can quickly spot changes in traffic behavior and don't take up much processing power. However, when handling intricate attack patterns, their detection accuracy is constrained

2) AI-Based Models: To increase detection accuracy, deep learning and sophisticated machine learning models are employed. The models listed below are tested and trained:

• **Multilayer Perceptron (MLP):** For organized traffic data, a conventional neural network model is appropriate.

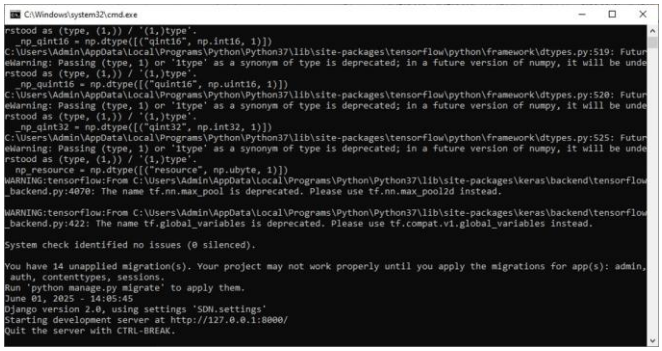

Fig. 4. Python command-line interface (CMD)

• **CNN1D:** Temporal dependencies in traffic flows are captured by a one-dimensional convolutional neural network.

• **CNN2D:** This article proposes an improved model that uses two-dimensional convolution to take advantage of spatial correlations between traffic features. Regarding assault identification, the CNN2D model surpasses CNN1D as it can catch complex feature interactions.

E. Model Evaluation and Selection

Trained models are evaluated using common performance criteria include accuracy, precision, recall, F1-score, and execution time. A comparative study is done to choose the best model for real-time detection. Based on the evaluation results, CNN2D is chosen as the last detection model because of its better accuracy and consistent performance.

F. Decision-Making Module

Once an assault is discovered, a decision-making module picks the best path of action for mitigation. If it is normal, traffic is allowed to flow across the network. The system rejects the matching packets if malevolent or DDoS activity is found in order to prevent congestion and service interruption. This automated response mechanism ensures rapid hazard response and minimizes possible damage.

G. System Deployment

MySQL is used for data storage while Python is used for logic execution and model building all across the framework. The system brings together model inference, dataset processing, and decision-making into one platform. This implementation enables thorough testing of the proposed technique in a regulated environment and illustrates how it can be applied to real SDN-based 5G networks.

V. Experimental Setup And Implementation

This section covers the precise means of implementation and experimental design used to confirm the proposed SDN-based security architecture for cloud-native 5G networks. The system is put in place in a controlled environment to see how well statistical and AI-based models for detecting attacks work.

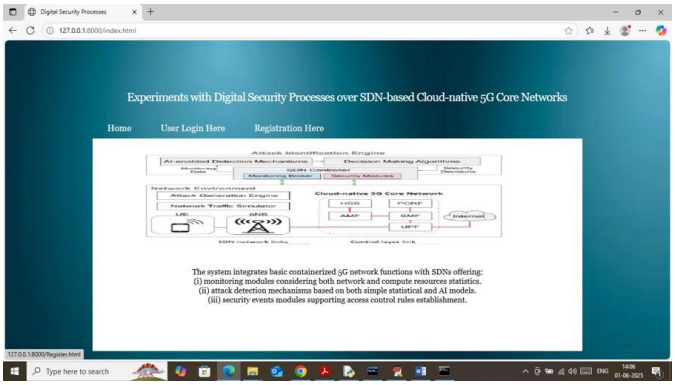

Fig. 5. Web-based user interface of the proposed SDN-enabled cloud-native 5G digital security system.

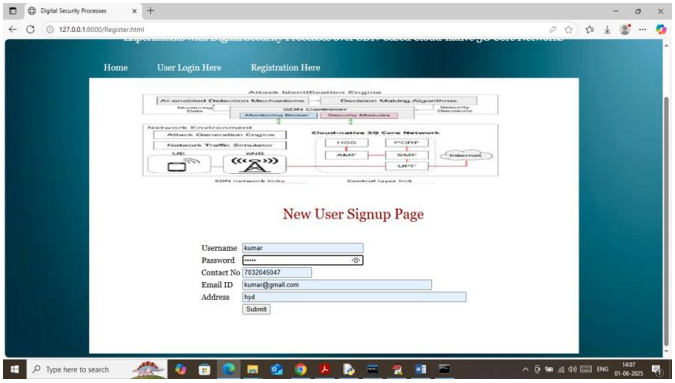

Fig. 6. User registration page of the developed web application.

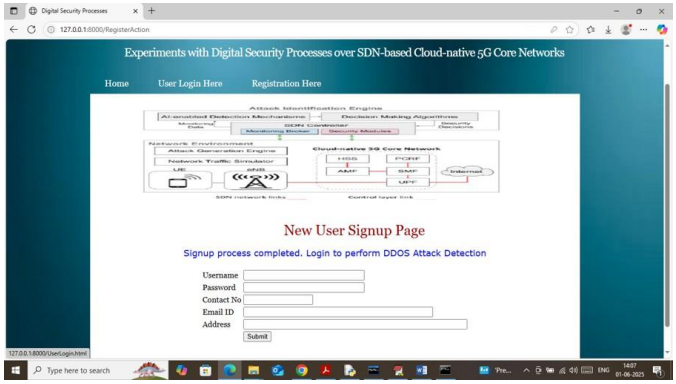

Fig. 7. User registration confirmation page of the web application.

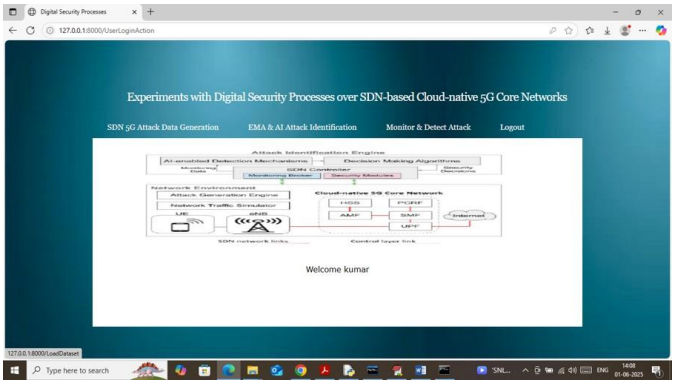

Fig. 8. User login success dashboard of the web application.

A. Software and System Configuration

The experimental setup is mostly created using Python (version 3.7.2). All required dependencies are installed using the packages mentioned in the requirements.txt file. MySQL database keeps model outcomes, user data, and dataset records. The database schema is set up using the included SQL scripts. To start the backend server, which controls user interface communication, model execution, and data processing, a Python-based web server is utilised. Once the server is working, a web browser lets you see the program nearby.

B. Application Workflow

The web-based interface provides limited system access via a user authentication procedure. Users have to log in to the program before using the detection modules. Following authentication, the later functional phases are executed.

C. Dataset Loading and Preparation

Loading the SDN–5G attack dataset requires the use of the application interface. The system shows, once loaded, important dataset information including class labels, feature count, and record count. Following that, the dataset is automatically divided into training and testing subsets; 20This phase makes sure that all the detection models process data the same way and that their performance is judged fairly.

D. Model Training and Evaluation

After the dataset is prepared, the training dataset is used to train statistical and AI-based models. The models below are used and judged: Exponential Moving Average (EMA) MLP, alternatively multilayer perceptron CNN1D is a one-dimensional convolutional neural network. CNN2D, two-dimensional convolutional neural network Trained models are used on the test dataset to determine performance criteria including accuracy, precision, recall, F1-score, and execution time. Comparative findings are shown graphically as well as in tables. The comparison of execution times draws attention to the trade-off between detection speed and accuracy. Statistical models run more faster, but AI models reach greater detection accuracy. Among all the models CNN2D comes out tops overall.

E. Attack Detection and Monitoring

Users can upload hitherto unknown SDN–5G traffic data to confirm real-time detection capability using the tool. Applying the best-performing model (CNN2D) to the given data classifies each traffic record as either a normal or DDoS attack. Alongside the categorization outcomes, the input traffic features and the expected labels are neatly presented. This approach shows the degree to which the system can detect harmful traffic and helps with automated network security decision-making.

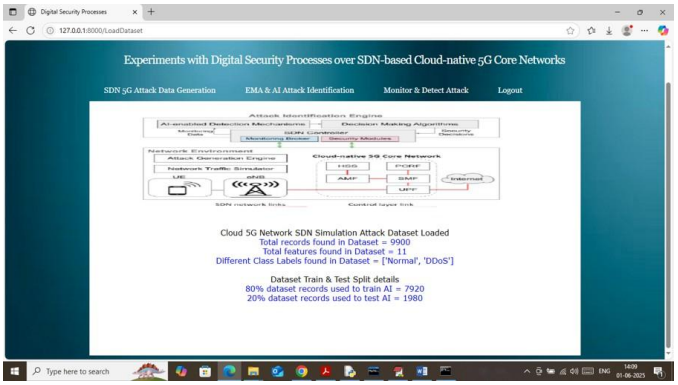

Fig. 9. Dataset loading and train–test split summary page of the web application.

Fig. 10. Performance comparison of attack detection algorithms.

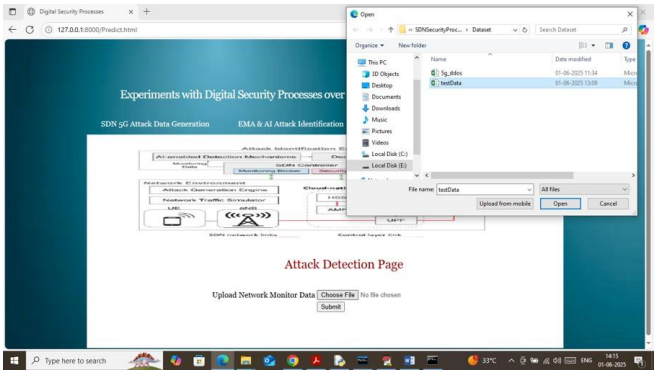

Fig. 11. Attack detection page with network data upload interface.

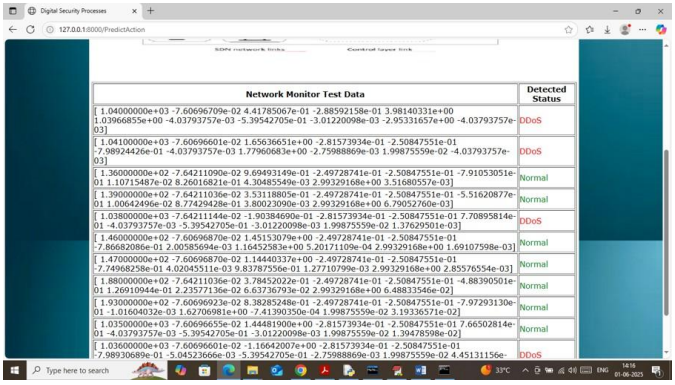

Fig. 12. Attack detection results showing normal and DDoS classifications.

F. Summary of Implementation

The experimental setup combines model training, evaluation, dataset processing, and attack detection into one venue rather well. The web-based version lets users easily interact with the proposed security architecture and shows how useful it is. According to the results, SDN-based AI models—particularly CNN2D—significantly increase detection accuracy and strengthen cloud-native 5G network security.

G. Performance Evaluation Metrics

VI. Results And Discussion

In this section, we go over the results of the tests on the proposed SDN-based cloud-native 5G security architecture and the performance of the detection models that have been put to use. The evaluation looks at detection accuracy, classification efficiency, and execution time to help decide if each model is good for real-time attack detection. The effectiveness of the detection systems is evaluated using standard performance metrics like accuracy, precision, recall, F1-score, and execution time. These metrics provide a reasonable assessment of detection quality and computational efficiency. While accuracy indicates how accurate the classification is overall, precision and recall demonstrate how well the model can identify malicious traffic. Execution time is analyzed to assess the viability of real-time deployment.

A. Comparative Analysis of Detection Models

The same training and testing datasets are used to compare the performance of AI-based models (MLP, CNN1D, and CNN2D) and statistical models (EMA and ARIMA). Due to their lightweight nature, statistical models execute quickly. However, when dealing with intricate and non-linear traffic patterns, their detection accuracy is comparatively low. Because of this drawback, they are less able to detect complex DDoS attacks in dynamic 5G environments. On the other hand, detection accuracy is much higher for AI-based models. The spatial and temporal dependencies present in network data are difficult for the MLP model to capture, despite its superior performance over statistical methods. When it comes to accuracy and recall, CNN1D outperforms MLP by identifying temporal patterns in sequential data. Out of all the techniques assessed, the CNN2D model performs the best overall. CNN2D efficiently captures spatial correlations between various traffic parameters using two-dimensional convolution, making it possible to identify attack patterns more precisely. Despite taking longer to execute than EMA and MLP, CNN2D is suitable for use in security-critical applications due to its higher accuracy.

B. Execution Time Analysis

Execution time analysis shows that there is a trade-off between speed and accuracy in detection. Deep learning

models take longer to process because they have complicated calculations. On the other hand, statistical methods finish detection quickly but are not very reliable. CNN2D has a little longer execution time than CNN1D among AI-based methods, although this gap is still acceptable for near real-time detection scenarios. These findings suggest that CNN2D can be used in SDN-controlled environments that provide automatic mitigation and centralized processing.

C. Attack Detection and Decision-Making Effectiveness

To verify real-time detection capacity, the trained CNN2D model is used on unseen SDN-5G traffic data. Traffic instances are successfully classified as either malicious or legitimate by the system. In order to avoid network congestion and service interruption, the decision-making module starts packet-dropping measures as soon as malicious traffic is identified. The efficacy of combining AI-based detection with SDN-enabled control is demonstrated by this automatic reaction system. The solution reduces the possible impact on network performance by properly identifying threats and responding quickly.

D. Discussion

The experimental findings verify that AI-based models perform noticeably better than conventional statistical techniques in identifying DDoS attacks in SDN-based 5G networks.CNN2D is particularly effective because it can evaluate complex feature correlations in network traffic. While statistical models are still useful for quickly identifying anomalies, they are insufficient to handle current 5G security concerns on their own.

VII. Conclusion And Future Work

This study presented an SDN-based security architecture for cloud-native 5G networks using statistical techniques and AI-driven models in order to properly identify and prevent attacks. The recommended approach combines intelligent analysis with centralized traffic monitoring to find bad activity in real time. Deep learning approaches beat traditional statistical methods in terms of detection accuracy, according to the research evaluating multiple detection models including EMA, ARIMA, MLP, CNN1D, and CNN2D. Among the models tested, CNN2D outpaced the others by effectively recording complex feature interactions in network traffic. Although statistical models offer faster execution, they are not suitable for handling sophisticated attack patterns in dynamic 5G environments since their limited detection capacity precludes them from managing them. By means of fast mitigation actions, such deleting harmful packets via SDN-based control, the inclusion of an automated decision-making module much strengthened the architecture. Based on the empirical findings, the proposed method can strengthen the dependability and resilience of the network against DDoS attacks. The framework could be improved by applying it in real-time 5G environments using real traffic and SDN hardware. To increase detection accuracy, one can look at other AI systems like hybrid deep learning models and Long Short-Term Memory (LSTM) networks. Enhancing the system also enables adaptive learning to manage multi-class attack classification and changing patterns of threat. One approach to get scalable and context-aware security in next-generation mobile networks is to combine the framework with network slicing and edge computing methods.

Acknowledgment

The authors would like to thank Ms. Y. Meghamala, Assistant Professor, Department of Electronics and Communication Engineering, Institute of Aeronautical Engineering, Hyderabad, for her great help, technical insights, and constant encouragement throughout the course of this research. Her knowledge and helpful criticism were very important in determining the direction and caliber of this work.

The authors also express their gratitude to the faculty members and staff of the Department of Electronics and Communication Engineering, Institute of Aeronautical Engineering, for giving the needed academic assistance, lab facilities, and computer tools for doing this study.

References

[1] I. Ahmad *et al.*, "Security for 5G and beyond," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 4, pp. 3682–3722, 2019.
[2] S. Kalafatidis and L. Mamatas, "Microservices-adaptive software-defined load balancing for 5G and beyond ecosystems," *IEEE Network*, vol. 36, no. 6, pp. 46–53, 2022.
[3] K. V. Cardoso *et al.*, "A softwarized perspective of the 5G networks," *arXiv preprint arXiv:2006.10409*, 2020.
[4] T. T. Nguyen and G. Armitage, "A survey of techniques for Internet traffic classification using machine learning," *IEEE Communications Surveys & Tutorials*, vol. 10, no. 4, pp. 56–76, 2008.
[5] N. Shone, T. N. Ngoc, V. D. Phai, and Q. Shi, "A deep learning approach to network intrusion detection," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 2, no. 1, pp. 41–50, 2018.
[6] R. Doshi, N. Apthorpe, and N. Feamster, "Machine learning DDoS detection for consumer Internet of Things devices," in *Proc. IEEE Security and Privacy Workshops*, pp. 29–35, 2018.
[7] "free5GC: Open-source 5G core network," [Online]. Available: https://www.free5gc.org/.
[8] "LABORA-INF-UFG NetSoft 2020 Tutorial 4 – Demo 2," [Online]. Available: https://github.com/LABORA-INF-UFG/NetSoft2020-Tutorial4-Demo2-Exp1. Accessed: Dec. 14, 2023.
[9] "Floodlight SDN Controller," [Online]. Available: https://github.com/floodlight/floodlight.
[10] N. McKeown *et al.*, "OpenFlow: Enabling innovation in campus networks," *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 2, pp. 69–74, 2008.
[11] G. Agrafiotis *et al.*, "Image-based neural network models for malware traffic classification using PCAP to picture conversion," in *Proc. 17th Int. Conf. on Availability, Reliability and Security*, pp. 1–7, 2022.
[12] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.
[13] "DoS/DDoS attacks on 5G networks dataset," [Online]. Available: https://www.kaggle.com/datasets/iagobs/dosddos-attacks-on-5g-networks.
[14] "6G-SANDBOX Malaga-UMA testbed," [Online]. Available: https://6g-sandbox.eu/pilot-6gsites/malaga/. Accessed: Dec. 14, 2023.