ISSN (Online):3048-5088

**International Journal of Advancement and Innovation in Technology and Research (IJAITR)**

**Volume 3, Issue 1, (Jan-Jun) 2025**

# DoS Attack Detection System Using Machine Learning

Radhika Ranawat
*Dept. of CSIT*
*Acropolis Institute of Technology and Research, Indore, India*

Yuvraj Pawar
*Dept. of CSIT*
*Acropolis Institute of Technology and Research, Indore, India*

Sonal Gupta
*Dept. of CSIT*
*Acropolis Institute of Technology and Research, Indore, India*
*sonalgupta230961@acropolis.in*

Niharika Soni
*Dept. of CSIT*
*Acropolis Institute of Technology and Research, Indore, India*
*niharikasoni230909@acropolis.in*

Vandana Kate
*Dept. of CSIT*
*Acropolis Institute of Technology and Research, Indore, India*
*vandanakate@acropolis.in*

Vinayak Sharma
*Dept. of CSIT*
*Acropolis Institute of Technology and Research, Indore, India*

***Abstract*—-New types of cyberattacks called Denial-of-Service (DoS) assaults now pose an extremely significant risk to contemporary computer networks, often leading to major disruptions in service availability and resulting in considerable economic damage through loss of productivity and revenue. Outdated system designs struggle to recognize new threats due to lack of adaptability; they frequently produce many unnecessary alerts. The study introduces an advanced system for detecting Distributed Denial of Service attacks through machine learning algorithms aimed at improving immediate cybersecurity measures in networks. A new mechanism intercepts real-time or archived internet data through an analyzer tool, analyzing it according to metrics like transmission frequency of packets, randomness in Internet Protocol headers, diversity among protocols used, and movements at specific ports. An algorithmic system utilizing supervision, namely a RandomForest classifier, undergoes training using annotated data sets for proficient identification of benign versus harmful network activity. A well-trained algorithm constantly watches for traffic on networks and sends notifications when it spots unusual behavior suggesting intrusion attempts. The experimental assessment reveals remarkable precision in detecting targets while significantly lowering error rates associated with incorrect identifications. This framework offers an adaptable, resourceful method for maintaining network security, guaranteeing continuous operation and reducing damage caused by changing digital risks.**

***Index Terms*—DoS Attack, Machine Learning, Cyber Security, Denial of Service.**

## I. Introduction

Crimes aimed at overwhelming targeted computer infrastructures via excessive traffic sent across multiple infected devices form part of DDoS attacks. Major disturbances at such an extent severely compromise online safety, convenience, and reliability substantially. A massive influx of requests across a system exhausts its bandwidth, disk space, and processing power, ultimately resulting in performance degradation or complete incapacitation. Recently, due to increased use of botnets for at-

ISSN (Online):3048-5088

International Journal of Advancement and Innovation in Technology and Research (IJAITR)

Volume 3, Issue 1, (Jan-Jun) 2025

tacks and automated methods, Distributed Denial Of Service (DDoS) assaults have occurred frequently and severely worldwide, affecting businesses, educational establishments, as well as individual users across various sectors. Many standard security tools such as firewalls and intrusion detection systems often fail at detecting new and evolving attacks. Therefore, researchers utilize advanced techniques like AI models involving machine learning and deep learning in order to detect and mitigate these threats. This approach evaluates network traffic information, detects crucial metrics, and classifies unusual behaviors more accurately and adaptively than conventional methods.

## II. Literature Review

The sophistication of DDoS attacks has increased dramatically due to evolving tactics employed by cybercriminals who craft increasingly sophisticated methods for overwhelming networks with traffic. Despite their effectiveness in identifying common threats, these signatures fall short when it comes to detecting novel or combined forms of cyberattacks. As a result of this development, ML and DL techniques now hold significant importance due to their capability in detecting new or changing types of attacks. Zeki et al[1]. The study examined various methods for detecting Distributed Denial-of-Service attacks within cloud infrastructure settings, highlighting issues related to managing multiple tenants and dynamic resource allocation adjustments. They suggested integrating both entropy-focused evaluation techniques and machine learning models like random forests and support vector machines in their approach towards efficient identification tasks. Research revealed that variations in IP addresses' randomness serve as precursors for detecting both coordinated assaults across networks and large-scale data breaches by anticipating preventive measures ahead of time.

Liu et al. A new approach combining principal component analysis and recurrent neural networks was presented in [2], aimed at detecting Distributed Denial of Service attacks. This method minimizes complex data dimensions through Principal Component Analysis (PCA) for enhanced computational speed; meanwhile, Recurrent Neural Networks (RNNs) identify sequential patterns within internet activity signals. An innovative approach showcased remarkable precision and F1 scores against conventional machine learning techniques, highlighting how integrating dimensionality reduction with time-series deep neural networks enhances system responsiveness while maintaining high fidelity in real-time applications.

The authors Al-Sharada et al. [3]'s detailed examination compared various methods for detecting Distributed Denial-of-Service attacks using machine learning and deep learning techniques against conventional classification algorithms. The results suggest that traditional machine learning techniques like decision trees and random forests continue to be effective because they offer better explainability and less demanding processing requirements compared to other methods. Nevertheless, deep learning models such as convolutional neural networks (CNNs), recurrent neural networks (RNNs) including Long Short-Term Memory units (LSTMs), and their hybrids excel at identifying complex attack vectors; however, these methods demand greater processing power and extensive training data compared to traditional approaches. Multiple persistent patterns surface in all of these investigations. Selecting features greatly affects both detection speed and overall system performance, especially when using combined approaches of initial screening by lightweight machine learning followed by detailed analysis through deep neural networks. Additionally, thorough assessment methods like temporal partitions in training sets versus testing, multiple dataset validations, and performance metrics tracking are highlighted as crucial for gauging practical applicability.

In conclusion, current studies indicate that forthcoming investigations ought to concentrate on creating adaptable and mixed methodologies that harmonise precision of identification, operational speed, and practicality for implementation within dispersed and online systems.

ISSN (Online):3048-5088

International Journal of Advancement and Innovation in Technology and Research (IJAITR)

Volume 3, Issue 1, (Jan-Jun) 2025

TABLE I: Summary of DDoS Detection Approaches

| Paper (year) | Data / Dataset | Main method(s) | Features used | Reported strengths | Limitations |
|---|---|---|---|---|---|
| Al-Shareeda et al. (2023) journal.beei.org | Survey of multiple works (no single dataset) | Comparative (ML vs DL) | Flow, statistical, entropy, time-window features | Good taxonomy, practical deployment notes | High-level survey (no new experiments) |
| Zekri et al. (2017) ResearchGate | Simulated / cloud traces | Entropy + ML classifiers (RF, SVM, etc.) | Entropy of src/dst IP/ports, packet/byte rates | Good fit for cloud / scalable flow monitoring; early detection focus | Limited to flow/statistical features; dataset realism depends on simulation |
| Q. Li et al. (PCA-RNN) (2019) SpringerLink | Real datasets (not all public details in preview) | PCA + RNN (temporal modelling) | Large feature set reduced by PCA; temporal windows | Lower inference time after PCA; RNN captures temporal bursts | Needs careful feature engineering; possible overfitting if PCA keeps too few components |

## III. Proposed Framework

A new framework integrates both level-based features detection alongside an ensemble of artificial intelligence techniques including neural networks for enhanced analysis capabilities. It comprises five phases: Data Collection: Benchmark datasets like CIC-DDoS-2019 and NSL-KDD can be accessed publicly. This collection includes annotated data of both benign network activity and malicious attacks characterized by various types such as UDP overloads, SYN storms, and HTTP assaults. The text is rephrased as follows: Preprocessing and Feature Engineering: The raw data of unprocessed internet communications, such as packet captures or NetFlow logs, undergoes aggregation at the level of individual flows for analysis purposes. Features statistically derived include: Packet rate, byte rate Flow duration Source/destination IP entropy Mean and variance of inter-arrival time Protocol and flag distributions Normalization techniques such as min-max scaling or z-score transformation are employed for feature adjustment. Hybrid Model Design: The algorithm combines conventional machine learning components with advanced neural network structures: An RF or SVM acts initially in classifying data into categories quickly for identifying anomalies efficiently. Flagged data is forwarded to a CNN-LSTM system which incorporates both spatiotemporal relationships for better identification of composite threats like vectorized or sporadic intrusions. Principal Component Analysis (PCA) combined with autoencoders helps reduce data dimensions while minimizing computation time and accelerating learning processes. Training and Validation: The data set has been partitioned into two subsets: an 70% portion for training purposes and a remaining 30% used exclusively for evaluation. The 5-fold cross-validation technique is employed to guarantee that models generalize well across different subsets of data. Models undergo training employing the Adam optimization algorithm at an initial learning pace set to 0. One thousand units along with a batch dimension set at sixty-four elements. Stopping early aims to avoid excessive model complexity. C. Evaluation Metrics Performance is evaluated using the following standard metrics: Accuracy (ACC) Precision (P) Recall (R) F1-score (F1) Detection Rate (DR) False Positive Rate (FPR) Processing Time (Tp) Our objective is to enhance accuracy by reducing both incorrect alarms and response times efficiently. D. Experimental Setup Research is carried out within an integrated platform utilizing tools like Scikit-learn, TensorFlow, and Keras. This device utilizes hardware featuring an Intel Core

ISSN (Online):3048-5088

International Journal of Advancement and Innovation in Technology and Research (IJAITR)

Volume 3, Issue 1, (Jan-Jun) 2025

i7 CPU, ample 16GB of RAM capacity, alongside integrated NVIDIA graphics capabilities for optimal performance. Comparisons of the proposed mixed-model's outcomes will be made relative to established benchmarks such as those involving decision trees, support vector machines, convolutional neural networks without additional layers, and recurrent neural network structures devoid of supplementary components. E. Expected Outcomes The hybrid RF + CNN-LSTM architecture is expected to surpass individual approaches regarding precision and resilience, especially when dealing with diverse and dynamic Distributed Denial-of-Service attack scenarios. Additionally, employing level-based statistical data enhances both performance and practicality when deploying systems on clouds or Internet of Things environments.

## IV. Proposed System / Methodology

This section describes the end-to-end system for network-traffic based intrusion/anomaly detection. The design covers data collection, feature extraction, model training, and the real-time detection alerting pipeline.

### A. Data Collection

Data collection is performed from two complementary sources to ensure diversity and realism.

*Live capture (Wireshark/PyShark):*

The system that is proposed will gather network traffic data by means of a combination of high-speed and programmable packet capture tools to guarantee both a good performance and flexibility. Among others, tcpdump and dumpcap will be used for high-speed packet capture and this will allow efficient recording of large amounts of the network with a minimum of packet loss. To do detailed and programmatic packet inspection, the system will use PyShark, a Python wrapper for Wireshark's TShark utility, to continuously extract and analyze packet-level information. The system allows for two capture modes: full-packet capture, which saves entire payloads for thorough analysis, and header-only capture, which greatly lowers the storage requirement by recording just the vital header details. The data comes from various strategic capture points, such as the network edge (router level), internal switch SPAN/mirror ports for observing lateral movement, and host-based agents for endpoint-level telemetry. Each packet that is captured will be augmented with metadata like NTP-synchronized timestamps, network interface identifiers, VLAN tags, applied capture filters, and session markers (start/stop), thus ensuring that the data is accurate, synchronized, and reproducible for further analysis and model training.

*PCAP datasets:*

The system to be proposed not only gets its training from live network captures but also uses non-private labelled PCAP datasets that are open to the public, to sufficiently cover the training data area and make it more diverse. As a result, the model is provided with a big learning area through the mixture of both bad and good traffic in these datasets. Besides, the normalization, filtering, and feature extraction that are all part of the preprocessing steps are documented in detail for reproducibility, clarity, and data integrity throughout the experimental process.

*Labelling:*

Supervised intrusion-detection research needs accurate labelling and ground truth that can be relied upon; in this research, labels are generated using a hybrid method of manual annotation, IDS-derived labelling, and controlled attack injection. Targeted manual annotation is done by domain experts for flows that are nuanced or ambiguous and for validating automated labels in focused experiments. To support this, we ingest the output of signature-based systems (e.g., Snort and Suricata) to automatically label flows that correspond to known malicious patterns, while keeping the original packet traces for auditing purposes. To guarantee full coverage of modern attack behaviours, we also conduct synthetic attack injection in a controlled testbed—replaying known exploit PCAPs and simulating adversaries with scripts—so that injected sessions are marked with precise start/stop times and canonical labels. Each label is accompanied by provenance metadata (label source, confidence

ISSN (Online):3048-5088
International Journal of Advancement and Innovation in Technology and Research (IJAITR)
Volume 3, Issue 1, (Jan-Jun) 2025

level, timestamp, and dataset version) and is kept together with the raw/processed data to ensure reproducibility, to analyse errors, and to permit conservative filtering or re-labelling during model evaluation.

*Storage versioning:*

In order to uphold data integrity and make possible the reproducibility of the results, the entire research process collects and processes data systematically according to a structured storage and versioning approach. The unprocessed PCAP files are kept in a secured object storage system that provides protection against data loss or corruption while at the same time allowing easy access and providing scalability. The flows that have been extracted and the feature sets are stored in different controlled dataset repositories in formats such as CSV or Parquet, which make efficient querying and analysis easier after preprocessing. Each dataset entry is linked to a complete metadata that comprises the dataset version, capture ID, label source, and preprocessing parameters, thereby guaranteeing the entire traceability from the original data through to the model-ready features. This systematic organization not only the experiment replication consistency support, but also makes retraining of the model easier and enables transparent comparison among different dataset versions or experimental configurations.

### *B. Feature Extraction*

From packets as well as flows we derive both low-level and aggregated features which are suitable for machine learning applications.

To analyze the traffic and compute the features efficiently, the system performs flow-level aggregation based on the standard 5-tuple — source IP, destination IP, source port, destination port, protocol — in addition to the flow start and end times. This aggregation combines single packets from the same communication session into one logical flow, hence data volume is reduced drastically while critical interaction patterns are still preserved. The system, by examining flows instead of raw packets, can produce significant statistics such as packet rate, byte count, and connection duration, which are extremely important for the detection of abnormal or malicious network behaviors.

*Example Features (Per-Flow / Per-Window)*

In order to depict the network behavior in a detailed manner, the system collects different types of features which include statistical, temporal, and behavioral ones at both flow and time-window levels. These features are categorized as follows:

**Traffic Volume & Rates:** Packet count, Byte count, Packets per second (pps), Bytes per second (bps)

**Timing Features:** Flow duration, Mean and standard deviation of inter-arrival times, Burstiness (variation in packet arrival patterns)

**Statistical Features:** Mean, median, and standard deviation of packet sizes, Payload entropy (randomness within data payloads)

**Behavioral Features:** Number of distinct destination IPs per source, Number of distinct destination ports per source, Average connection attempts per minute

**Protocol/Flag Features:** TCP flag counts (SYN, FIN, RST), Ratio of SYN to ACK packets, UDP/TCP traffic ratio

**Entropy & Distributional Features:** Source IP entropy, Destination IP entropy, Port usage entropy (helps detect scanning behavior)

**Derived Features:** Ratio-based metrics (e.g., bytes/packets), Rolling-window deltas (change in rate compared to baseline)

These feature groups collectively enable the detection system to identify a wide range of anomalies and attack patterns, from subtle traffic deviations to large-scale distributed scans.

### *C. Feature Engineering Normalization*

In order to keep the performance of the model at the same level and to make the learning process more efficient, the system employs a number of preprocessing procedures, which include feature engineering and normalization techniques: 1. Log Transformation: Heavy-tailed numeric features, like packet counts and byte volumes, are log-transformed so that their distributions become less

ISSN (Online):3048-5088

International Journal of Advancement and Innovation in Technology and Research (IJAITR)

Volume 3, Issue 1, (Jan-Jun) 2025

skewed and their variances more uniform. 2. Normalization: Using either a Robust Scaler (to manage outliers) or Min-Max Scaling (to restrict values to a certain range), all the features are normalized so that they are equally important during model training. 3. Categorical Encoding: Non-numeric features like protocol types or service labels are turned into numeric form with one-hot encoding or target encoding, depending on the number of unique values in the features and the size of the dataset. 4. Feature Selection: To lower dimensionality and make the model more interpretable, mutual information, permutation importance, and correlation study are used to detect and discard redundant or non-informative features. These preprocessing methods support the formation of a feature set that is clean, well-scaled, and rich in information, making it possible for the model to learn efficient and accurate intrusion detection patterns.

### D. Windowing Strategy

The system uses a windowing strategy for data aggregation and feature computation in order to efficiently recognize the temporal patterns and short-term variations in network behavior. Fixed-length sliding windows (e.g., 10 seconds, 60 seconds) are utilized to segment traffic continuously over time, which allows for the detection of sudden spikes or anomalies in packet rates and connection activity. Along with that, adaptive session windows based on flow end times are employed allowing a dynamic aggregation that coincides with actual communication sessions instead of arbitrary time intervals. The hybrid method guarantees that both time-driven and session-driven traffic characteristics are precisely represented, thereby enhancing the system's capability to detect real-time intrusions and evolving attack behaviors significantly.

## V. Model Training

The recommended method applies a supervised learning technique for intrusion detection, prioritizing model reliability, validation, and interpretability. RandomForestClassifier is selected as the foundational model because of its resistance against feature scaling, excellent precision, and capacity to render understandable feature importance values. Moreover, sophisticated models including XGBoost, LightGBM, and sequence-based deep learning frameworks like LSTM or CNN could be tested to determine network traffic flows' intricate temporal dependencies.

### A. Training Pipeline

The dataset gets allocated into different sets of training, validation, and testing through a time-aware split that secures against the leakage of temporal data, thus capturing and testing the model's performance on future data, respectively, thereby simulating deployment in real life. The stratified time-block method is applied to cross-validation which guarantees that there is no class imbalance in the folds and chronological order is observed at the same time. Hyperparameter optimization is done through grid search or randomized search, adjusting the key parameters like `n_estimators`, `max_depth`, and `min_samples_leaf` according to the validation outcome.

### B. Class Imbalance Handling

In order to balance the classes of benign and attacks, the RandomForest algorithm applies the model with class weights. Other tactics consist of SMOTE-driven oversampling used only for the training set and adjusting the detection threshold to cut down on the number of false positives during the detection phase.

### C. Evaluation Metrics

- **Major metrics:** The attack class Precision, Recall, and F1-score, stressing reliable detection plus negligible false alarms.
- **Operational metrics:** ROC-AUC, PR-AUC, and detection latency (which is the time from attack commencement to detection).
- **Deployment metrics:** False positive rate per hour and Mean Time to Detect (MTTD) are the measures that will help to guarantee that the deployment is efficient in real-world conditions.

ISSN (Online):3048-5088
International Journal of Advancement and Innovation in Technology and Research (IJAITR)
Volume 3, Issue 1, (Jan-Jun) 2025

### D. Explainability & Validation

The model's decisions are interpreted with the help of tools such as SHAP and permutation importance, and it is verified that the feature's significance corresponds with the knowledge of the network's domain. Testing for robustness is carried out by adding noise, slight changes in the payload, or obfuscation attempts all aimed at assessing model's resilience to adversarial behavior.

### E. Model Lifecycle Management

Model Lifecycle Management involves the logging of each trained model with comprehensive metadata such as the training date, version of the dataset, hyperparameters, and the performance on validation. The model undergoes periodic retraining or is retrained whenever concept drift is detected through the monitoring of feature distributions changes or detection accuracy over time.

The systematic training approach assures that the model is always dependable, clear, and capable of adapting to the changes in the network conditions and attack patterns that are taking place.

## VI. Detection & Alerting (Real-Time Pipeline)

The detection and alert notification system are the operational heart of the suggested IDS (Intrusion Detection System), which allows for the immediate surveillance, evaluation, and counteraction of harmful activities over the network. The pipeline consists of uninterrupted packet capture, real-time model inference, smart alerting, and visual display for security researchers.

### A. Runtime Capture & Preprocessing

Packet data in real-time is gathered by means of lightweight agents like dpkt, Scapy, or PyShark who are working in live-capture mode. Moreover, the system has the capability to receive data from streaming telemetry sources such as NetFlow or IPFIX, which are designed for high-throughput flow extraction, as well as very low latency. The processed traffic is then sent through the same step of the preprocessing pipeline that is applied during model training to guarantee consistency in feature representation. Normalization, window-based aggregation, and transformation of the features are included in this process; thus, the system can keep the same data characteristics of training and live environments.

### B. Real-Time Classifier

The RandomForest model that has been trained is not only available as a RESTful microservice but also run as a part of a stream processing framework, like Apache Flink or Kafka Streams, for continuous inference. The model is serialized with joblib for fast loading and low-latency predictions which also provides the advantage of scalability and efficiency. In very busy situations, you can use either a worker pool or a light ensemble made up of combining rule-based filters and ML classifiers to be able to manage throughput and accuracy at the same time.

This strategy guarantees that the system is capable of handling large scale and real-time traffic without affecting the performance much.

### C. Decision Logic and Thresholds

So as not to lose the ability to detect the real cases while at the same time not having too many false alarms, the classification thresholds are regulated very closely saw from validation results. Some aggregation rules can be used like needing several detections within a very short time interval or ensemble voting before an alert is sent out. This multi-layered logic not only helps to cut off the passing anomalies but also directs the attention to the consistent and very confident detections.

### D. Alerting & Response

When suspicious activity is detected, the system triggers automated alerts through various communication and monitoring channels, including:

- SIEM integration (e.g., Elastic Stack, Graylog, Splunk)
- Email/SMS webhooks for instant notifications
- Automated ticketing systems for workflow management

Each alert includes a contextual payload containing:

- Affected host(s) and timestamp
- Detected attack type or anomaly class

ISSN (Online):3048-5088
International Journal of Advancement and Innovation in Technology and Research (IJAITR)
Volume 3, Issue 1, (Jan-Jun) 2025

- Confidence score and key contributing features (based on SHAP analysis)
- Recommended mitigation or response actions

This structured alert information allows network administrators to respond quickly and effectively to potential threats.

### *E. Dashboard & Analyst Tools*

A real-time monitoring dashboard provides visual insights into network activity and system performance. It displays:

- Live alerts and their severity levels
- Time-series trends of traffic metrics and model outputs
- Top influential features contributing to each detection (from SHAP values)
- Analyst playbooks for guided triage and response procedures

This interface enhances situational awareness and simplifies the process of investigating and verifying alerts.

### *F. Fail-Safes & Performance Monitoring*

To ensure reliability under all conditions, the system includes multiple fail-safes and monitoring mechanisms:

- **Graceful degradation:** If the ML model service becomes unavailable, the system automatically falls back to a signature-based IDS for baseline protection.
- **Performance tracking:** Inference latency, throughput, and alert rates are continuously monitored.
- **Model drift detection:** Feature distributions and prediction trends are logged and analyzed periodically to identify drift, prompting retraining when necessary.

### *G. Privacy & Legal Considerations*

The system adheres to strict data privacy and compliance standards. It logs only essential metadata and minimizes the storage of sensitive packet data, ensuring user anonymity. Access control policies are enforced on raw captures, and a well-defined data retention policy is implemented in accordance with institutional and regulatory guidelines.

This real-time detection and alerting pipeline ensure a seamless transition from data capture to actionable intelligence, enabling rapid threat detection, contextual awareness, and adaptive security responses in dynamic network environments.

## VII. Conclusion and Future Work

This research work has developed a system for intrusion detection and alerting in real-time which is capable of spotting possible network anomalies thanks to its intelligent flow-level analysis and the use of machine learning. The system brings together live packet capturing, feature extraction, and a RandomForest-based classification model to provide detection of network threats that is efficient and with very low latency. A structured data pipeline is used for the whole process—comprised of preprocessing, windowing, and alert generation—so that the model can adjust to changing traffic patterns, and be scalable as well as interpretable at the same time. The most important advantages of this system comprise detection in real time, flow aggregation that is adaptive, alerts that are explainable through visualization of feature importance, and compatibility with existing security infrastructures like SIEMs. Its modular structure permits flexible deployment both in enterprise and cloud environments, thus providing strong protection without incurring much in terms of computational overhead. Improvements can be made in the future work. One of the improvements could be the use of deep learning architectures such as LSTMs or Transformers, which would not only help in better recognizing temporal patterns but also in getting rid of manual feature engineering completely. The extension of the model to multi-class attack classification would provide the capability of detecting different types of threats in a finer way. Further, the combination of online learning for model drift adaptation and the use of federated training will not only maintain data privacy but also guarantee continuous improvement of the model.

ISSN (Online):3048-5088

**International Journal of Advancement and Innovation in Technology and Research (IJAITR)**

**Volume 3, Issue 1, (Jan-Jun) 2025**

## REFERENCES

[1] M. Zekri, S. E. A. El-Kafrawy, and N. Aboutabit, "DDoS Attack Detection Using Machine Learning in Cloud Computing Environments," in *IEEE CloudTech*, 2017.

[2] Q. Li et al., "A PCA-RNN Based DDoS Attack Detection Model," in *Advances in Intelligent Systems and Computing*, Springer, 2019.

[3] A. Al-Shareeda et al., "A Survey on Machine Learning-Based DDoS Attack Detection Techniques," *Bulletin of Electrical Engineering and Informatics*, vol. 11, no. 1, pp. 432–445, 2022.

[4] Kate, V., Shukla, P. (2021). A 3 Tier CNN model with deep discriminative feature extraction for discovering malignant growth in multi-scale histopathology images. Informatics in Medicine Unlocked, 24, 100616.

[5] V. Kate, A. Jaiswal and A. Gehlot, "A survey on distributed deadlock and distributed algorithms to detect and resolve deadlock," 2016 Symposium on Colossal Data Analysis and Networking (CDAN), Indore, India, 2016, pp. 1-6, doi: 10.1109/CDAN.2016.7570873.

[6] V. Kate, R. Ushasree, R. M. Tharsanee, M. T. Kukreja, S. Saraf and B. Varadharajan, "GAN, CNN and ELM Based Breast Cancer Detection," 2023 2nd International Conference for Innovation in Technology (INOCON), Bangalore, India, 2023, pp. 1-6, doi: 10.1109/INOCON57975.2023.10101250.