IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 10, October 2026

DOI: 10.17148/IARJSET.2026.131063

Automated Detection and Mitigation of DDoS Attacks and Flash Crowds Using Machine Learning for High-Traffic Web Servers: A Review of Automated Cyber Security Attack Analysis

S. K. Chougule[1], Prathamesh Choudhary[2], Payal Ghadage[3], Satyajit Jadhav[4], Aditi Mhaske[5]

Department of Computer Engineering, PDEA's College of Engineering, Manjari (Bk.), Pune, India[1-5]

Abstract: When many genuine users open a web service at once, for example when examination results are announced, the server can overload in the same way as under a distributed denial of service (DDoS) attack. The two events need opposite responses, yet both raise request volume sharply. This paper reviews twelve recent studies, together with earlier work they build on, covering machine learning for intrusion, DDoS, malware and ransomware detection. It examines how they analyze traffic and logs, reconstruct attack sequences and support automated defense. A comparison by objective, method, evidence and limitation shows that most systems stop at detection, are evaluated on a single dataset, and rarely connect detection to mitigation. Only one reviewed study models attack lifecycles, and none of the reviewed studies, as far as we could verify, treats a genuine flash crowd as a class separate from attack traffic. Based on these gaps, we outline a research direction: a traffic classifier that separates normal, flash crowd and DDoS traffic and drives rate limiting, queuing and blocking automatically. No experimental results are claimed.

Keywords: cyber security, attack analysis, intrusion detection, DDoS, flash crowd, machine learning, automated mitigation, attack lifecycle

I. INTRODUCTION

When a popular service announces something that many people are waiting for, such as examination results, a very large number of genuine users open the same web page within seconds. The server slows down or fails and users see errors, although no attacker is involved. The same symptoms appear when a DDoS attack floods the server with requests. The first event is a flash crowd and the second is an attack. Both raise request volume sharply, but the correct responses are opposite: genuine users should still be served, possibly with limits and queuing, whereas attacking sources should be blocked. A system that cannot separate the two will either block real users or let the attack through.

Automated attack analysis turns raw evidence into decisions: whether an attack is present, which technique is in use, how far it has progressed and which response is appropriate. The evidence is spread across network traffic, system logs and host statistics [1], and attacks unfold as sequences of techniques, which the MITRE ATT&CK framework organizes into tactics, techniques and lifecycles [1], [2]. Machine learning (ML) is now the dominant approach to detection, but published systems differ widely in what they analyze, how they are evaluated and whether they act on what they find.

This paper reviews twelve recent studies [1], [9]–[19], selected for a final-year project on automated attack analysis, together with the earlier work on which they build [3]–[8]. The review (i) summarizes how existing studies analyze traffic and logs, reconstruct attack sequences and support automated defense; (ii) compares them in a common table; (iii) identifies research gaps, notably the lack of flash-crowd handling and of closed-loop mitigation; and (iv) outlines a research direction that is clearly separated from the reviewed work. The paper is a literature review. The authors have not yet carried out experiments, and no experimental results are claimed.

II. BACKGROUND AND RELATED WORK

A. Flash crowds and DDoS attacks

A flash crowd is a sudden rise in legitimate demand, whereas a DDoS attack is malicious traffic from many sources intended to exhaust the server. Both increase the request rate, so a simple volume threshold is unreliable. The behavior

© IARJSET This work is licensed under a Creative Commons Attribution 4.0 International License

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 10, October 2026

DOI: 10.17148/IARJSET.2026.131063

of each source can instead be described by features such as its request rate, the interval between requests, the pages requested and the packet size. These are the features considered in the proposed direction (Section VI).

B. Data sources and the analysis pipeline

Network flows expose the number of packets, ports, protocol and duration; system logs record the activity of the operating system and the software running on it; and host statistics describe each process, for example CPU use, memory use and page-fault rate [1]. Network-based attacks leave traces mainly in traffic data, malware infections in system logs, and cryptocurrency mining is better detected through network and host statistics together [1]. Reviewed pipelines typically perform one-hot encoding of categorical features, cleaning, removal of duplicated features and scaling, handle class imbalance with oversampling such as SMOTE, train one or more classifiers and report the F1 score, which balances precision and recall [1].

C. Techniques and attack lifecycles

In ATT&CK, a tactic is the objective of one stage of an attack and a technique is the means used to achieve it; an attack lifecycle is the ordered sequence of techniques used in a whole attack [1], [2]. Recognizing the lifecycle, rather than only the presence of an attack, helps a security operations center choose countermeasures that prevent the attack from completing, and can guide the design of detectors that act at earlier stages [1].

D. Earlier ML-based detection

As characterized in [1], earlier studies can be grouped into attack-level, technique-level and lifecycle-level detection, and most work is at the attack level. On network traffic, Zavrak and İskefiyeli [3] used autoencoder variants on CICIDS2017 and NSL-KDD, and Toupas *et al.* [4] used a deep neural network for multi-class classification with resampling to counter class imbalance. On system logs, DeepLog [5] and LogAnomaly [6] model log sequences with LSTM networks. Lin *et al.* [7] combined packet flows, system logs and host statistics with XGBoost, but performed binary attack detection. At the technique level, Bagui *et al.* [8] built a network dataset labeled by reconnaissance and discovery tactics and compared ML algorithms, without modeling lifecycles. These studies detect whether and what an attack is, but give limited insight into its objectives, targets and progression [1].

III. LITERATURE REVIEW OF SELECTED STUDIES

A. Attack lifecycle and multi-source analysis

Lin *et al.* [1] move from attack-level to lifecycle-level detection. They compare a single-stage ML model, a two-stage ML+ML pipeline and ML with edit-distance sequence matching (ML+SM) on CREMEv2, a testbed-generated dataset with five attack scenarios (Mirai botnet, ransomware, resource hijacking, disk wipe and endpoint denial of service) built from 17 ATT&CK techniques, with network traffic, system logs and host statistics collected at one-second granularity. In the first stage of the two-stage design, one model per data source detects techniques (XGBoost for traffic and host statistics, LSTM for logs) and an ensemble merges the outputs by soft voting with a threshold on the benign class. In the second stage, a Bi-LSTM classifies the resulting technique sequence into a lifecycle.

The two-stage design reached an F1 score of 0.994 for lifecycle detection, against 0.887 for the single-stage model and 0.189 for ML+SM. The single-stage model classified only benign data well and struggled because techniques are shared between lifecycles; ML+SM failed because the opening techniques of ransomware, resource hijacking and disk wipe are identical, and because it depends on exact technique recognition. For technique detection, the ensemble of all three sources reached an F1 of 0.922, against 0.877 for traffic alone and 0.578 for logs alone, the latter attributed to noisy log labeling. Every ensemble containing more than one source outperformed each of its individual sources. The authors acknowledge the burden of synchronizing sources, privacy, storage and real-time processing, the extra computation of ensembles, and the limited number of lifecycles, which required data augmentation of the sequences.

B. DDoS detection

Pebrianto and Suryani [9] detect DDoS attacks with entropy, a dynamic threshold and re-evaluation of suspicious IP addresses. The adaptive threshold reduces manual tuning, although false positives remain possible, and the scope is DDoS detection rather than broader multi-attack analysis. Fotse *et al.* [10] apply federated learning to DDoS detection (FedLAD), mainly in software-defined networking (SDN) environments, which limits its applicability to other attack types. A comparative study of DDoS detection on 5G-SDN [11] likewise focuses on one network environment. None of these summaries describes a separate class for legitimate traffic surges.

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 10, October 2026

DOI: 10.17148/IARJSET.2026.131063

C. Network intrusion detection with deep and ensemble models

Picot *et al.* [12] apply a deep neural network to packet-based, protocol-independent attack detection and report strong accuracy, precision, recall and F1; the approach needs large, representative traffic data and its robustness to manipulated packets is not sufficiently evaluated. Ali *et al.* [13] combine a BERT transformer, a multilayer perceptron and SMOTE and reach up to 99.39% accuracy, yet the computational demand limits deployment on low-power edge devices. A two-level ensemble framework [14] improves detection and generalization over single models at the price of added complexity. Work on balanced multi-class detection [15] addresses class imbalance but is dataset-dependent and provides no automated mitigation after detection. A comparison of several classifiers for cyber-attack monitoring [16] mainly separates attack from normal traffic.

D. Malware, ransomware and messaging fraud

Panja *et al.* [17] use Random Forest and Extra Trees with feature influence for malware detection (99.39% accuracy); the approach works on static data and may miss fileless malware residing in memory. Ispahany *et al.* [18] propose iCNN-LSTM+, combining CNN, LSTM and attention for incremental ransomware detection (99.65% F2), but it depends on Sysmon log formats. A study of an AI-based SMS filtering system [19] reports that most of its 110 respondents were highly satisfied with its reliability, and credits real-time filtering and continuous learning; it is limited to one operator and to SMS, and is the least related to network attack analysis.

TABLE I COMPARISON OF THE REVIEWED STUDIES

Ref., year	Objective	Method	Key finding	Advantage	Limitation	Research gap
[1], 2024	Detect attack techniques and lifecycles from several data sources	Single-stage ML, ML+ML, ML+SM; ensemble of traffic, log and host models	F1 0.994 (lifecycle, ML+ML); F1 0.922 (technique)	Three data sources; recovers the attack lifecycle	Needs several synchronized sources; higher complexity	Synchronization, real-time processing, scalability
[9], 2025	DDoS attack detection	Entropy, dynamic threshold, suspicious-IP re-evaluation	Improved detection	Adaptive threshold; less manual tuning	Possible false positives	DDoS only, not broader attack analysis
[10], 2025	DDoS detection, mainly in SDN	Federated learning (FedLAD)	–	–	–	Mainly SDN; limited to DDoS
[11], –	Comparative DDoS detection on 5G-SDN	Comparison of detection approaches	–	–	–	Limited to 5G-SDN and DDoS
[12], 2025	Protocol-independent network attack detection	Deep neural network on packet data	Strong accuracy, F1, precision, recall	Protocol-independent	Needs large, representative traffic data	Adversarial robustness not sufficiently evaluated
[13], 2024	Network intrusion detection	BERT + MLP + SMOTE	Up to 99.39% accuracy	High accuracy on intrusion patterns	Depends on dataset and training quality	High compute; limits edge deployment
[14], 2024	Network intrusion detection	Two-level ensemble learning	Improved detection performance	Better generalization than single models	More complex than single models	Not specified
[15], 2024	Multi-class intrusion detection	ML with class balancing	Improved multi-class detection	Handles class imbalance	Dataset-dependent performance	No automated mitigation after detection

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 10, October 2026

DOI: 10.17148/IARJSET.2026.131063

Ref., year	Objective	Method	Key finding	Advantage	Limitation	Research gap
[16], 2024	Cyber-attack monitoring and detection	Several ML classifiers	Compared by accuracy, precision, recall, F1	Compares several methods	Mainly attack-vs-normal classification	Not specified
[17], 2025	Malware detection	Random Forest + Extra Trees, feature influence	99.39% accuracy	Efficient, feature-based	Depends on feature quality	Static data; may miss fileless malware
[18], 2025	Incremental ransomware detection	CNN + LSTM + attention	99.65% F2	Spatial and sequential features	Higher computational needs	Depends on Sysmon log format
[19], 2026	Spam and fraud SMS detection	AI-based SMS filtering	Majority of 110 respondents highly satisfied	Real-time; continuously learns	Limited to one operator and SMS	Limited to one operator, city and SMS

Values are as reported by each study on its own data and are not directly comparable. Entries are limited to information verified from the study summaries; a dash means not available.

IV. COMPARATIVE ANALYSIS

A. Methods and data

Method choice follows data type (Table I): deep and transformer models for packet and network data [12], [13], CNN-LSTM for logs [18], tree ensembles for tabular features [1], [17], and entropy statistics for DDoS [9]. In [1], tree-based models, especially XGBoost, were strongest on traffic and host statistics (F1 of 0.877 and 0.724), while recurrent models were the weakest component on system logs. Tree-based models also trained in minutes, for example about 6.8 minutes for XGBoost on traffic, whereas recurrent models needed roughly 105 to 196 minutes on logs [1].

B. Evaluation

Headline scores of about 99% [13], [17], [18] come from different datasets and tasks and cannot be compared directly; [1] itself states that a direct comparison with earlier work is infeasible because objectives and methodologies differ. Metrics also differ: [18] reports the F2 score, which weights recall more heavily than precision, whereas [1] uses F1. Class imbalance is handled by oversampling in [1], [4] and [13], and by dedicated balancing in [15]. For a web-server setting, the reviewed studies do not report the rate at which genuine users are wrongly blocked, although [9] notes that false positives are possible.

C. Sequence reasoning and deployment cost

Only [1] reasons over sequences of events, and its result shows that a learned second stage tolerates imperfect technique predictions better than rigid pattern matching: the learned stage reached 0.994 while edit-distance matching reached 0.189. The cost of such multi-stage, multi-source designs is real. In [1], testing 6000 data points took about 94 seconds for the single-stage model and about 102 seconds for ML+SM, and heavier models in [13] and [18] raise computational requirements, which matters for real-time protection of a loaded server.

V. RESEARCH GAPS AND CHALLENGES

G1. Flash crowd versus DDoS. Both produce high request volume, yet none of the reviewed studies, as far as we could verify, treats a genuine surge as a separate class, and [9] itself flags possible false positives. *G2. Detection without response.* Only [15] explicitly notes the absence of automated mitigation, and the verified information shows no reviewed study evaluating a closed loop from detection to blocking or rate limiting. *G3. Dataset and environment dependence.* Results in [1], [13] and [15] rest on specific datasets, and [9]–[11] are limited to DDoS or to SDN environments.

G4. Limited sequence analysis. Lifecycle reasoning appears only in [1], which covers five lifecycles. *G5. Cost and real-time operation.* Multi-source analysis requires synchronization and computation [1], and ensembles and large models add complexity [13], [14]. *G6. Data quality and robustness.* Noisy labels reduced log-based detection in [1], adversarial manipulation is insufficiently evaluated [12], and [1] leaves unknown attacks and adversarial resilience as future work.

© IARJSET This work is licensed under a Creative Commons Attribution 4.0 International License

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 10, October 2026

DOI: 10.17148/IARJSET.2026.131063

VI. PROPOSED RESEARCH DIRECTION

The following is a plan and not a completed result. The direction targets G1, G2 and G3 for high-traffic web servers, motivated by result-announcement portals where a flash crowd and a DDoS attack can look alike. Three research questions guide it. RQ1: can per-source behavioral features separate normal, flash crowd and DDoS traffic with a low false-positive rate? RQ2: can a decision engine keep genuine users served and the server stable while attackers are blocked? RQ3: which models give the best balance of accuracy and cost?

A. Architecture

Fig. 1 shows the planned pipeline. Web server traffic is captured from Nginx logs and packet data; Python converts it into per-IP features such as request rate, request intervals, requested pages and packet size; an ML classifier labels traffic as normal, flash crowd or DDoS; a decision engine maps the label to an action; and mitigation is applied automatically. A report summarizes requests served, sources blocked, false positives and server stability, and Prometheus and Grafana provide monitoring.

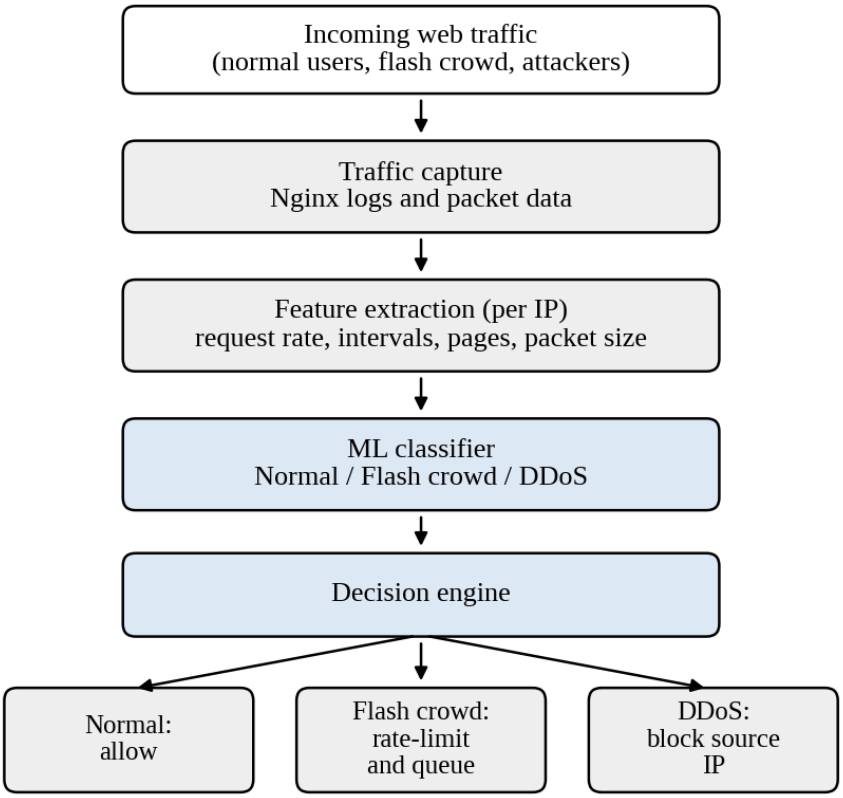


Fig. 1. Planned analysis and mitigation pipeline (proposed design).

B. Decision policy

Table II gives the policy. The distinguishing choice, relative to the reviewed systems, is that a flash crowd is never blocked: its users are rate-limited and queued so that everyone is eventually served.

TABLE II PLANNED DECISION POLICY

Class	Action	Mechanism
Normal	Allow	Pass to server
Flash crowd	Rate-limit and queue; block nobody	Nginx rate limiting
DDoS	Block source IP	iptables

© IARJSET This work is licensed under a Creative Commons Attribution 4.0 International License

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 10, October 2026

DOI: 10.17148/IARJSET.2026.131063

C. Data, tools and evaluation

Training data would combine the CICIDS2017 dataset [20] with normal, flash-crowd and attack traffic generated in an isolated Docker laboratory, using Locust for normal and flash-crowd load and hping3 for attack traffic against an Nginx server, so that results do not rest on one dataset. Random Forest, XGBoost and SVM would be compared using scikit-learn [21]. Evaluation would report accuracy, precision, recall, F1, false-positive rate and server stability before and after mitigation, in line with the metrics used in [1]. The planned design targets are detection within 1.5 seconds and precision above 95%; these are targets and not results. Attack traffic would be generated only inside the authors' own laboratory, never against a real college or public server.

D. Relation to existing work

The intended contribution is the flash-crowd class and the closed loop from classification to mitigation, not a new classifier. The reviewed systems stop at detection [15], are limited to DDoS or specific networks [9]–[11], or analyze other attack families [17], [18]. The two-stage idea of [1] is a natural later extension, as are the future directions below. Main risks are that flash-crowd and DDoS traffic can look similar, so false positives must be measured carefully, and that synthetic traffic may not reflect real events.

VII. FUTURE SCOPE

Planned extensions include LSTM or other deep learning models for zero-day attack detection; Layer 7 attacks such as HTTP flood and Slowloris; a CAPTCHA or proof-of-work challenge for suspicious users; auto-scaling and CDN or load-balancer integration on a cloud platform; real-time streaming with Kafka for very high request volumes; and explainable AI such as SHAP to show why an address was blocked.

VIII. CONCLUSION

The reviewed studies show strong ML detection scores but fragmentary attack analysis: sequence-level reasoning appears in one study, response automation is generally out of scope, and flash crowds are not separated from attacks. The evidence in [1] suggests that context and multiple data sources improve accuracy, while its reported synchronization and computation costs argue for lightweight features in a web-server setting. The proposed direction, a classifier that separates normal, flash crowd and DDoS traffic and drives automatic mitigation, addresses these gaps. Its implementation and measurement remain future work.

REFERENCES

[1]. Y.-D. Lin *et al.*, "Two-stage multi-datasource machine learning for attack technique and lifecycle detection," *Comput. Secur.*, vol. 142, Art. no. 103859, 2024, doi: 10.1016/j.cose.2024.103859.

[2]. B. E. Strom, A. Applebaum, D. P. Miller, K. C. Nickels, A. G. Pennington, and C. B. Thomas, "MITRE ATT&CK: Design and philosophy," The MITRE Corporation, Tech. Rep., 2018.

[3]. S. Zavrak and M. İskefiyeli, "Anomaly-based intrusion detection from network flow features using variational autoencoder," *IEEE Access*, vol. 8, pp. 108346–108358, 2020.

[4]. P. Toupas, D. Chamou, K. M. Giannoutakis, A. Drosou, and D. Tzovaras, "An intrusion detection system for multi-class classification based on deep neural networks," in *Proc. 18th IEEE Int. Conf. Mach. Learn. Appl. (ICMLA)*, 2019, pp. 1253–1258.

[5]. M. Du, F. Li, G. Zheng, and V. Srikumar, "DeepLog: Anomaly detection and diagnosis from system logs through deep learning," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur. (CCS)*, 2017, pp. 1285–1298, doi: 10.1145/3133956.3134015.

[6]. W. Meng *et al.*, "LogAnomaly: Unsupervised detection of sequential and quantitative anomalies in unstructured logs," in *Proc. IJCAI*, vol. 19, no. 7, 2019, pp. 4739–4745.

[7]. Y.-D. Lin, Z.-Y. Wang, P.-C. Lin, V.-L. Nguyen, R.-H. Hwang, and Y.-C. Lai, "Multi-datasource machine learning in intrusion detection: Packet flows, system logs and host statistics," *J. Inf. Secur. Appl.*, vol. 68, Art. no. 103248, 2022.

[8]. S. Bagui *et al.*, "Detecting reconnaissance and discovery tactics from the MITRE ATT&CK framework in Zeek Conn logs using Spark's machine learning in the big data framework," *Sensors*, vol. 22, no. 20, Art. no. 7999, 2022.

[9]. Pebrianto and Suryani, "Adaptive DDoS attack detection: Entropy-based model with dynamic threshold and suspicious IP reevaluation," 2025.

[10]. Fotse *et al.*, "Federated learning based DDoS attack detection (FedLAD)," 2025.

[11]. "Comparative detection of DDoS attacks on 5G-SDN."

© IARJSET This work is licensed under a Creative Commons Attribution 4.0 International License

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 10, October 2026

DOI: 10.17148/IARJSET.2026.131063

[12]. Picot *et al.*, "Protocol-agnostic packet-based intrusion detection," 2025.
[13]. Ali *et al.*, "BERT transformer + MLP for intrusion detection," 2024.
[14]. "A two-level ensemble learning framework for enhancing network intrusion detection systems," 2024.
[15]. "Balanced multi-class network intrusion detection using machine learning," 2024.
[16]. "Cyber-attack monitoring and detection using machine learning techniques," 2024.
[17]. Panja *et al.*, "Efficient malware detection using feature influence," 2025.
[18]. Ispahany *et al.*, "iCNN-LSTM+: Incremental ransomware detection," 2025.
[19]. "Airtel's AI SMS filtering system: A user perception towards spam protection," 2026.
[20]. I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," in *Proc. Int. Conf. Inf. Syst. Secur. Privacy (ICISSP)*, 2018.
[21]. F. Pedregosa *et al.*, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, 2011.

© IARJSET This work is licensed under a Creative Commons Attribution 4.0 International License