

Adaptive Anomaly Detection of CTS Flooding in Wi-Fi Networks Using Sketch-Based Profiling

Asif Hossain Joy, Zinadin Zidan, Nahar Sultana

Department of Computer Science, Faculty of Science and Technology,

American International University - Bangladesh, Dhaka, Bangladesh

Email: 22-47850-2@student.aiub.edu, 22-47918-2@student.aiub.edu, nahar@aiub.edu

Abstract—Using Wi-fi in both public and private networks could be challenged by IEEE 802.11 MAC-layer protocol problems. This study concentrates on the Clear-to-Send (CTS) flood attack, characterized as a low-effort, high-impact threat that is significantly perilous and not widely recognized. This vulnerability “poisons” the Network Allocation Vector (NAV) of real devices by taking advantage of the fact that CTS control packets do not need to be checked. This weakness makes them silent for long periods of time and badly messes up communication. Today’s intrusion detection systems (IDS) do not know about this issue since they focus on monitoring data and management frames. In addition, traditional rule-based protections also do not anymore work, which may be easily avoided by attackers. The proposed system learns the typical time, frequency, and length of the CTS traffic behavior, allowing it to identify anomalous changes. This paper offers a novel approach to detecting abnormalities employing sketch-based profiling to address significant security vulnerabilities. The proposed architecture offers a flexible and scalable solution to safeguard various Wi-Fi networks, ranging from public networks to corporate campuses, against an unidentified attack vector.

Index Terms—CTS flooding, Anomaly Detection, Wi-Fi, Sketch-based Profiling

I. INTRODUCTION

Adaptive anomaly detection is a dynamic security approach that keeps learning from normal network behavior and identifies any activity that may look like malicious activity, adjusting its detection mechanisms over time to handle evolving attack patterns. In Wi-Fi networks, Clear-to-Send (CTS) frames are control packets that are used in the IEEE 802.11 MAC protocol to manage channel access and prevent collisions, but they can be exploited by attackers to monopolize the wireless medium through flooding attacks. For addressing the problem, sketch-based profiling uses lightweight data structures such as Count-Min Sketch and Bloom filters to efficiently summarize and track CTS traffic patterns without storing complete packet histories, enabling real-time detection with low computational overhead. Wireless networks have become the central part of today’s modern communication, enabling devices, sensors, and smart systems to interact seamlessly across a wide range of environments [1]. The cable-free design facilitates deployment, mobility, and easy expansion in homes, enterprises, and public spaces, which has been repeatedly emphasized as a defining advantage of wireless technologies [2]. Using radio-based communication, these networks enable real-time data sharing, remote access, and simultaneous connectivity of numerous

devices, forming the backbone of today’s digital and IoT ecosystems [3]. While allowing broad participation of nodes, it simultaneously demands stable performance and reliable coordination for efficient channel use [4]. The proposed Algorithm 1 implements a multi-stage detection framework that combines sketch-based profiling with PHY-layer verification and challenge-response mechanisms to identify and mitigate CTS flooding attacks. As wireless communication continues to expand into critical applications, ensuring that devices can access the medium safely becomes increasingly important, especially because misuse of control mechanisms, such as manipulation of CTS frames, can directly undermine fair channel availability [5].

The key challenge in securing wireless communication happens when control-frame procedures are misused to influence medium access, particularly through manipulation of the CTS function that governs NAV settings. Studies examining the impact of duration fields on channel reservation indicate that even minor adjustments can compel adjacent nodes to endure unnecessary waiting periods, thereby diminishing channel availability and impairing throughput [6]. Investigations into CTS duration misuse in practical WLAN environments similarly show that such alterations distort access behavior without being noticed when underlying timing parameters are not cross-checked [1]. The difficulty was further reinforced when analyzing the MAC-layer vulnerabilities, which highlight that the conventional IDS implementations rarely compare CTS behavior across layers or inspect subtle inconsistencies in reservation timing [4]. The issue becomes detectable when examining recognizable patterns such as irregular NAV-reuse cycles, unusual sender activity trends, and shifts in PHY-layer characteristics. When we combine lightweight verification techniques, these cross-layer indicators can provide a reliable basis for distinguishing legitimate CTS activity from suspicious manipulation, aligning with broader insights from cross-layer IDS methodologies.

Several research have looked at how RTS/CTS modification affects access to wireless media. They have found that changing duration fields or making up control frames can make channels less available and make communication less effective [4]. Experimental studies have shown that it is possible to use RTS/CTS based interference on real 802.11 networks and have pointed out the flaws in MAC-layer reservation systems. Testbed studies of several types of attacks showed that inflated

reservation values make throughput go down and wait times go up for valid nodes [5]. To stop this form of exploitation, some have recommended using duration checking and NAV-validation methods. These plans can discover reservation times that don't match and keep the medium from being used for too long. More research has been done on manipulating CTS, and methods like duration re-evaluation and address comparison have been shown to improve latency and throughput after being fixed [1]. Complementary cross layer techniques have also examined the detection of counterfeit CTS through PHY/MAC correlation. These investigations demonstrate that integrating address verification with physical-layer indicators facilitates the identification of suspect control frames [7]. The following is a summary of this work's main contributions:

- Introduces a lightweight uniformity-based detection framework that replaces rigid rule checks and heavy ML models with fast statistical mechanisms, enabling real-time CTS monitoring with minimal overhead.
- Combines Count-Min Sketching, Bloom-filter NAV pattern analysis, and PHY-layer fingerprinting to detect abnormal CTS behavior, providing a multi-evidence detection approach not used in prior CTS defense works.
- Implements an on-demand challenge response verification step to confirm suspicious transmitters before any action is taken, preventing false positives and addressing spoofing attacks ignored by traditional solutions.
- Applies adaptive soft mitigation including token-bucket rate limiting and NAV rewriting to control attacker impact without disrupting legitimate users, avoiding the blind blacklisting and broadcast alerts seen in earlier research.
- Conducts a controlled simulation-based experimental evaluation using Google Colab to validate the logical effectiveness of the proposed detection and mitigation algorithm, demonstrating CTS flooding detection and impact reduction under modeled traffic conditions without relying on real packet-level data structures.

Uses agreement between different sniffers and adjusts thresholds to keep accuracy steady in various network situations, providing flexible and self-adjusting security for today's Wi-Fi environments.

II. STATE OF ART WORKS

Studies investigating vulnerabilities in IEEE 802.11 MAC operations have consistently demonstrated that the manipulation of RTS/CTS signaling modifies media access and undermines channel fairness [6]. Fundamental analyses in [8] [6] revealed that slight distortions in duration fields directly affect contention cycles, compelling compliant nodes into needlessly extended NAV states. Subsequent calculations in [2] and empirical assessment in [5] validate that fabricated or inflated reservation intervals can markedly diminish WLAN throughput. These works demonstrate that the misuse of CTS length is both possible and consequential, particularly when IDS models depend exclusively on MAC-level header validation [1].

In [8], a subtle timing inconsistencies propagate through the network. Flow-level monitoring in [9] and cooperative relay behavior observed in [10] illustrated that timing irregularities in control frames can cascade into queue instability and unpredictable link-layer performance. In [1] and [4] noted that CTS-duration alteration is particularly difficult to spot when systems fail to correlate NAV patterns with PHY-layer characteristics. Furthermore, as highlighted in [11], even minor timing misalignment's during overlapping transmissions can mislead PHY-layer energy detection and MAC-layer back-off processes, allowing hidden-device interactions to silently escalate into persistent channel-access disruptions.

Lightweight IDS frameworks in [3] and selective mitigation approaches in [12] demonstrated the benefit of integrating statistical MAC indicators with machine-learning models to isolate abnormal timing patterns. Analytical evaluations of control-frame geometry in [13] and multi-link scheduling studies in [14] emphasized how accurate RTS/CTS exchange is vital for interference coordination, particularly in dense multi-link or multi-AP environments. These studies show that while CTS irregularities generate identifiable behavioral signatures, many existing frameworks either impose high overhead, require heavy computation, or lack real-time responsiveness [1].

In [15] highlights a repeated reservation misuse creates virtual jamming conditions, where nodes defer access despite no legitimate transmitter occupying by the medium. Similarly, the performance-driven observations in [8] showed that stability deteriorates rapidly when timing inconsistencies exceed protocol tolerance levels. Analyses in [6] shows that artificially extended RTS/CTS durations force neighboring nodes into prolonged NAV states, showing how reservation misuse propagates deterministic timing distortions throughout the network. In [4] identifies duration-field manipulation as a primary driver of repeatable congestion and channel blocking, linking these timing deviations to broader fairness and performance degradation.

III. SYSTEM MODEL

Imagine a busy café where customers are connected to the Wi-Fi through several access points placed around the room. Most users are browsing, studying, or working online. In the same space, an attacker quietly uses a laptop to send a series of forged CTS frames toward one of the café's access points. These fake CTS signals tell nearby devices that the wireless channel will stay busy for longer than normal, forcing them to stop transmitting. As more of these forged frames reach the access point, some of the users experience slow connections, delayed uploads, and interruptions. The network appears active, but legitimate devices struggle to communicate because the attacker keeps extending the channel reservation unfairly.

Our model begins working as soon as these CTS frames start reaching the network. In the first stage, the system collects all incoming Wi-Fi frames but pays special attention to CTS messages. It observes who is sending them, how often they appear, the NAV value they contain, and the basic radio signal

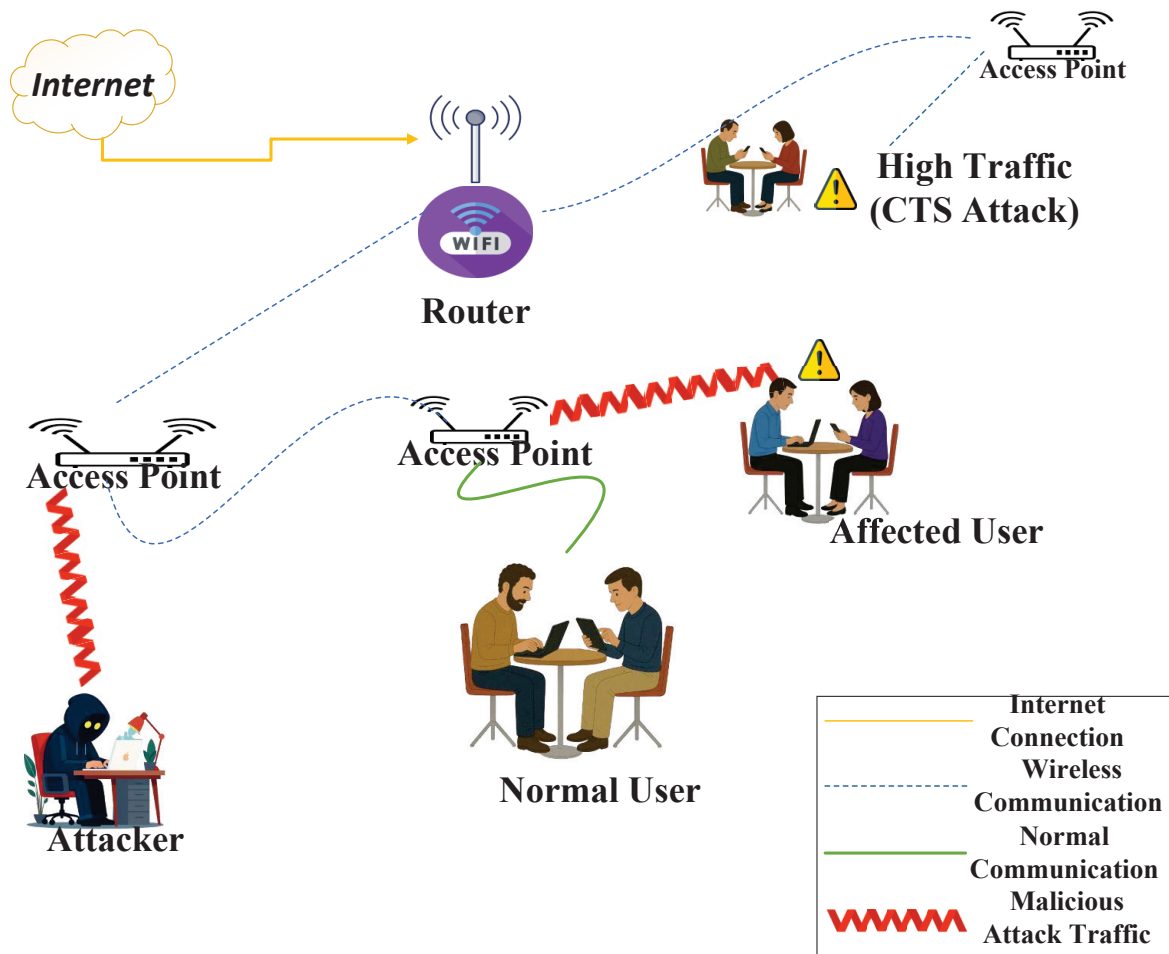


Fig. 1: Illustration of a CTS flood attack impacting Wi-Fi users

features behind them. The system quickly learns what regular CTS activity looks like with these small bits of information, and it can immediately tell when something strange happens. If the CTS behavior seems out of place or too frequent for a single device, the system marks the traffic as suspicious and moves to deeper analysis.

In the second stage, the system examines how the overall traffic flow changes after suspicious CTS packets appear. It compares the new behavior with the café's usual wireless activity and if the CTS pattern becomes too repetitive, arrives too quickly, or does not match the physical signal pattern of a legitimate sender, the system becomes confident that this is not normal usage. For avoiding any mistakes, it performs an additional confirmation step, challenging the suspicious sender and checking whether the response matches what a genuine client would produce. If the sender fails this check, the system confirms that the CTS frames are malicious.

Once an attack is verified, the final stage activates to protect the users. Instead of instantly blocking devices, the system applies soft and controlled actions. It can restrict how often the attacker's CTS frames can influence the channel

and correct any harmful NAV values so that other users do not remain silent unnecessarily. Even after these measures if an attackers continue, the system will gradually escalates to stronger actions, such as isolating the device or notifying the network administrator. Over time, the system learns from the traffic conditions and adjusts its internal configuration to handle more advance or subtle attacks in the future.

IV. PROPOSED METHOD

The Algorithm in Algo. 1 depicts the details of searching CTS attack in the neighborhood covered with Wi-fi network. Through maintaining several process and extraction after three stages offers an novel approach ensuring reliable wi-fi networks by adopting abnormal activities.

A. Operational Architecture

In the proposed system the Algo. 1 is designed to identify abnormal CTS activity through fast and lightweight processing. The model operates in three stages. In the data preprocessing stage, each incoming CTS frame is examined with minimal feature extraction, capturing the sender address, NAV field, timestamp, and a compact PHY-layer signature. These features

are summarized using Count-Min Sketch, Bloom-filter NAV tracking, and a short PHY-history buffer, enabling the system to maintain an efficient behavioral profile for every device. In the traffic evaluation stage, this algorithm performs sequential anomaly scoring, where CTS transmission rates, NAV-reuse behavior, and PHY-layer variations are analyzed together.

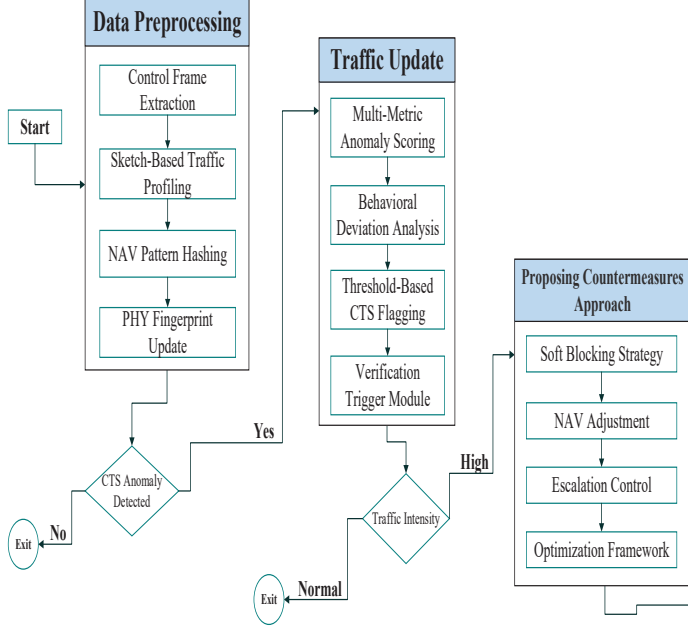


Fig. 2: Block diagram of the proposed system

If these measurements exceed their expected range, the system activates an on-demand challenge–response mechanism to confirm the authenticity of the suspicious sender. In the final mitigation stage, the system applies token-bucket rate limiting, NAV rewriting, and cross-sniffer consensus to contain the attacker while ensuring that legitimate users are not affected. This three-stage pipeline offers low computational cost, clear interoperability, and reliable performance in real-time Wi-Fi environments.

B. Key Word Extraction

1) *Count-Min Sketch*: **Scenario**: When a device starts sending CTS frames at an unusual rate in the network. **Action**: Sketch quickly estimates the sender's CTS frequency and identifies heavy senders without storing full packet details.

2) *Bloom-Filter NAV Tracking*: **Scenario**: When an attacker repeatedly injects the same NAV values to keep other devices silent. **Action**: Bloom filter detects repeated NAV patterns and flags suspicious NAV-reuse behavior instantly.

3) *PHY-Layer Signature History*: **Scenario**: When a malicious node spoofs someone else's MAC address to hide its identity. **Action**: PHY history compares signal characteristics to detect mismatches, revealing spoofed transmitters.

4) *Sequential Anomaly Scoring*: **Scenario**: When combined CTS features begin deviating from the normal traffic

Algorithm 1: Identification of Malicious CTS Sources Using Profiling and Verification

Input: f : incoming CTS frames; CM, B_{NAV}, PHY : counter map, NAV-pattern buffer, PHY-history set
Output: A : detected malicious CTS senders; Mitigation: actions against offenders

```

1:  $C \leftarrow \emptyset$ ;  $A \leftarrow \emptyset$ 
2: COLLECT_CTS_FEATURES()
3: for each frame  $f$  do
4:   if  $type(f) = CTS$  then
5:      $(s, d, n, t, r, i) \leftarrow extract(f)$ 
6:     UPDATE_SKETCH( $s$ )
7:     UPDATE_NAV_PATTERN( $s, n, d$ )
8:     UPDATE_PHY_SET( $s, r, i$ )
9:     if  $CM[s] > R_{sk}$  then
10:       $C \leftarrow C \cup \{s\}$ 
11:   end if
12: end if
13: end for
14: for each sender  $s \in C$  do
15:    $\hat{c} \leftarrow norm(CM[s])$ 
16:    $\hat{n} \leftarrow reuse(B_{NAV}, s)$ 
17:    $\hat{p} \leftarrow dist(PHY[s])$ 
18:   if  $(\hat{c} > Th_c)$  and  $(\hat{n} > Th_n \text{ or } \hat{p} > Th_p)$  then
19:      $x \leftarrow rand()$ ; SEND_CHAL( $s, x$ )
20:     if  $resp(s, x, T)$  and  $match(PHY[s])$  then
21:       $C \leftarrow C \setminus \{s\}$ 
22:   else
23:      $A \leftarrow A \cup \{s\}$ 
24:   end if
25: end if
26: end for
27: for each  $s \in A$  do
28:   APPLY_RATE( $s$ ); NORM_NAV( $s$ )
29:   if  $median(scores(s)) > Th_{cons}$  then
30:     ALERT( $s$ )
31:   end if
32:   if  $persist(s)$  then
33:     BLOCK( $s$ )
34:   end if
35: end for
36: TUNE_PARAMETERS(); UPD( $R_{sk}, Th_n, Th_p, Th_c$ )
37: return  $A$ 
  
```

pattern. **Action**: Scores changes in rate, NAV behavior, and PHY drift to determine if CTS activity is anomalous.

5) *Challenge–Response Verification*: **Scenario**: When CTS traffic looks suspicious, but confirmation is required before taking action. **Action**: System sends a random challenge; only a legitimate associated device can respond correctly, confirming authenticity.

6) *Token-Bucket Rate Limiting*: **Scenario**: When a confirmed malicious node continues sending high CTS bursts. **Action**: The system limits how frequently that node's CTS

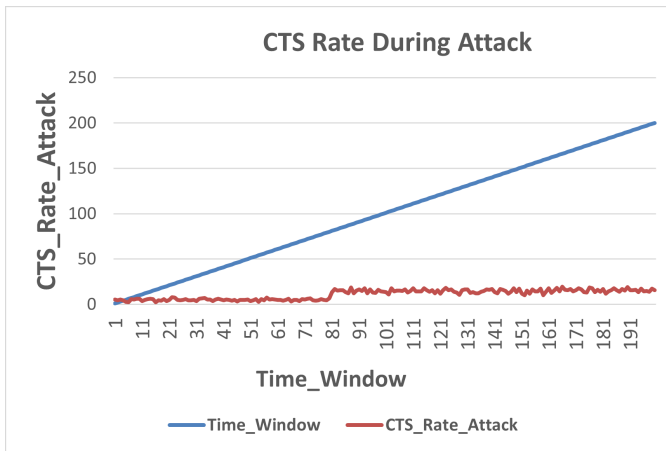


Fig. 3: CTS rate during attack

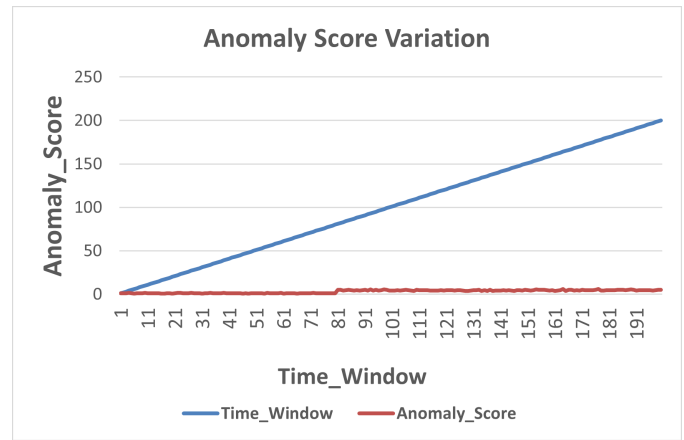


Fig. 5: Anomaly score variation

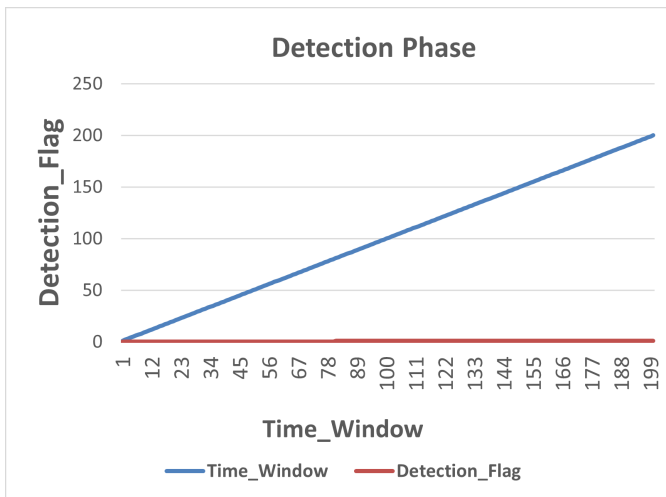


Fig. 4: Detection phase

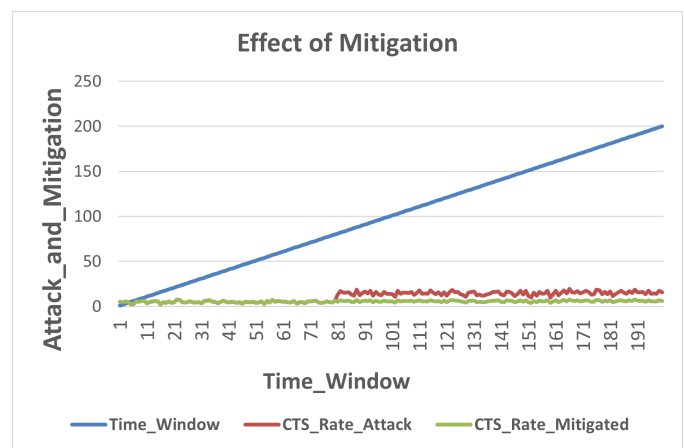


Fig. 6: Effect of mitigation

frames can influence the medium.

7) *NAV Rewriting*: **Scenario**: When the attacker's CTS frames attempt to extend NAV values to silence the network. **Action**: AP rewrites the NAV to a safe value so legitimate users are no longer blocked.

8) *Cross-Sniffer Consensus*: **Scenario**: When one sniffer detects abnormal CTS behavior and needs confirmation from others. **Action**: Multiple sniffers share small anomaly scores; only if the median exceeds a threshold is the attacker marked fully confirmed.

C. CTS Flooding Detection and Mitigation Results

Fig. 3 illustrates the CTS transmission rate over time. During normal network operation, the CTS rate remains low, indicating stable behavior. Once the attack begins, a sharp increase in the CTS transmission rate is observed, revealing CTS flooding behavior. This demonstrates that abnormal CTS activity can be effectively identified by monitoring the CTS transmission frequency.

Fig. 4 presents the detection output of the proposed algorithm. Initially, no attack is detected and the detection flag

remains zero. When the CTS flooding activity exceeds the predefined thresholds, the detection flag switches to one, indicating malicious behavior. This result shows that the proposed algorithm can detect CTS flooding attacks shortly after their onset.

Fig. 5 shows the anomaly score calculated using CTS transmission rate, NAV reuse, and PHY deviation. Under normal operating conditions, the anomaly score remains low. After the attack begins, the score increases and exceeds the detection threshold, signaling anomalous activity. These results confirm that combining multiple CTS-related features improves the accuracy and robustness of CTS flooding attack detection.

Fig. 6 illustrates the CTS transmission rate before and after the application of the mitigation mechanism. Prior to mitigation, the CTS rate remains elevated due to the flooding attack. Once the attack is detected, the proposed mitigation strategy significantly reduces the CTS transmission rate. This demonstrates that the mitigation mechanism effectively limits the impact of CTS flooding while preserving normal network communication.

D. Novelty and Comparison

This algorithm differs sharply from earlier CTS defense methods. Prior works [5] [1] depend on duration re-evaluation, RA/TA matching, and direct blacklisting, which are sensitive to spoofing and generate unnecessary broadcast traffic. Machine-learning IDS systems [3] need heavily supervised classifiers and large feature sets, making them unsuitable for unauthenticated control frames. The survey [4] shows that existing defenses struggle with low-rate CTS manipulation and rarely incorporate physical-layer checks. Our approach introduces a combination of probabilistic data structures, multi-indicator anomaly scoring, PHY-layer verification, and adaptive mitigation which creates a more reliable and resource-efficient detection path. By validating suspicious behavior only after multiple lightweight indicators agree, the system minimizes false positives and avoids premature blocking. These design improvements make the proposed algorithm a more scalable, precise, and practical solution for enhancing the security of the 802.11 control-frame layer compared to existing IDS approaches.

V. CONCLUSION AND FUTURE WORK

Since the proposed model has not been experimentally validated in real wireless environments, this work first conducted a controlled simulation-based experimental evaluation to analyze the logical effectiveness of the proposed CTS flooding detection and mitigation algorithm. The simulation was performed by emulating CTS traffic behavior according to the proposed algorithm's decision logic, allowing observation of CTS rate variation, anomaly detection response, and mitigation impact under modeled network conditions. The results demonstrate that the proposed algorithm can successfully detect abnormal CTS flooding behavior and reduce its effect through soft mitigation, thereby supporting the feasibility of the proposed approach at the logical and behavioral level. Future work will focus on building a controlled Wi-Fi test environment to evaluate the model's performance in real conditions. For the implementation phase need tools such as Scapy for generating CTS traffic patterns, Aircrack-ng/AirMon-ng for frame injection and monitoring, and Wireshark for detailed packet capture and analysis. To simulate network conditions more realistically, Mininet-WiFi or a small Raspberry Pi-based testbed can be used to reproduce different CTS intensities, varying client densities, and mobility scenarios. This setup will allow us to measure how accurately the Count-Min Sketch, Bloom-filter NAV tracking, and PHY-layer verification respond under real signal variations. The challenge-response mechanism will also be validated to ensure legitimate devices are always distinguished from spoofed nodes. After confirming stability in controlled tests, the model will be deployed in a live network environment to assess scalability, false positive behavior, and real-time responsiveness. Additional improvements can be done including tuning adaptive thresholds through lightweight ML libraries such as scikit-learn and adding enhanced PHY-level features for stronger validation. This experimental stage

will be crucial for showing that the suggested method works in practice and is ready to be used.

REFERENCES

- [1] Shariq Butt, Tauseef Jamal, Aneela Zameer, Pedro Amaral, Asifullah Khan, and Kiramat Kiramat. Denial of service attack in wireless lan. In *Proceedings of the Twelfth International Conference on Digital Society and Governments (ICDS 2018)*. IARIA, 2018.
- [2] Tauseef Jamal. Design and performance of wireless cooperative relaying. Technical report, Lusófona University, 2013.
- [3] Amar Amouri, Vishwa T. Alaparthi, and Salvatore D. Morgera. A machine learning based intrusion detection system for mobile internet of things. *Sensors*, 20(2):461, 2020.
- [4] Supriya S. Sawwashere and Sonali U. Nimbhorkar. Survey of rts-cts attacks in wireless network. In *2014 Fourth International Conference on Communication Systems and Network Technologies*, pages 752–755. IEEE, 2014.
- [5] Tauseef Jamal, Muhammad Alam, and Muhammad Mussadiq Umair. Detection and prevention against rts attacks in wireless lans. In *2017 International Conference on Communication, Computing and Digital Systems (C-CODE)*, pages 152–156. IEEE, 2017.
- [6] P M D Nagarjun, V. Anil Kumar, Ch Aswani Kumar, and Ahkshaey Ravi. Simulation and analysis of rts/cts dos attack variants in 802.11 networks. In *2013 International Conference on Pattern Recognition, Informatics and Mobile Engineering*, pages 258–263. IEEE, 2013.
- [7] Xiaocheng Zou and Jing Deng. Detection of fabricated cts packet attacks in wireless lans. In *Quality, Reliability, Security and Robustness in Heterogeneous Networks*, pages 105–115. Springer, 2012.
- [8] Lin Dai and Xinghua Sun. A unified analysis of ieee 802.11 dcf networks: Stability, throughput, and delay. *IEEE Transactions on Mobile Computing*, 12(8):1558–1572, 2013.
- [9] Shui Yu, Wanlei Zhou, Weijia Jia, Song Guo, Yong Xiang, and Feilong Tang. Discriminating ddos attacks from flash crowds using flow correlation coefficient. *IEEE Transactions on Parallel and Distributed Systems*, 23(6):1073–1080, 2012.
- [10] Tauseef Jamal, Paulo Mendes, and André Zúquete. Wireless cooperative relaying based on opportunistic relay selection. *International Journal On Advances in Networks and Services*, 5, 2012.
- [11] Hsueh-Wen Tseng, Shan-Chi Yang, Ping-Cheng Yeh, and Ai-Chun Pang. A cross-layer scheme for solving hidden device problem in ieee 802.15.4 wireless sensor networks. *IEEE Sensors Journal*, 11(2):493–504, 2011.
- [12] Soyoung Joo, So-Hyun Park, Hye-Yeon Shim, Ye-Sol Oh, and Il-Gu Lee. Machine learning-based detection and selective mitigation of denial-of-service attacks in wireless sensor networks. *Computers, Materials Continua*, 82:2475–2494, 2025.
- [13] Yi Zhong, Zhuoling Chen, Wenyi Zhang, and Martin Haenggi. Dual-zone hard-core model for rts/cts handshake analysis in wlans. *IEEE Transactions on Wireless Communications*, 24(11):9323–9337, 2025.
- [14] Zhuoling Chen, Yi Zhong, and Martin Haenggi. Cross-link rts/cts for mlo mm-wave wlans. 2025. Early Access.
- [15] Dazhi Chen, Jing Deng, and P.K. Varshney. Protecting wireless networks against a denial of service attack based on virtual jamming. In *Proceedings of the ACM Workshop on Wireless Security (WiSe 2003)*. ACM, 2003.