iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** VII **Month of publication:** July 2026

DOI: https://doi.org/10.22214/ijraset.2026.84032

www.ijraset.com

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

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue VII July 2026- Available at www.ijraset.com

From Black Box to Firewall: Automating Snort Rules from SHAP Explanations in IoT DDoS Detection

Aayush Rajbhar[1], Sunny Nahar[2]
[1]*MCA Student, Vivekananda Education Society's Institute of Technology (VESIT)*
[2]*Mentor, Vivekananda Education Society's Institute of Technology (VESIT)*

Abstract: ***With the explosive growth of Internet of Things (IoT) hardware, securing modern networks has become increasingly difficult as they are often the launchpad for massive Distributed Denial of Service (DDoS) campaigns. Security teams have been successful in deploying Machine Learning (ML) classifiers like Random Forests to detect these threats. However, there is a fundamental problem: these models provide zero transparency. Network administrators are not keen to deploy defensive measures based solely on an opaque algorithm's prediction. Explainable AI (XAI) methods such as SHapley Additive Explanations (SHAP) are able to provide the reasoning behind these models, but at present, they only generate visual graphs for humans to interpret. In this work we go beyond visualisation and develop a fully autonomous defence system. We developed an algorithm that takes local SHAP data and programmatically builds it into functional Snort 3 firewall commands immediately. Our Random Forest engine was able to identify DDoS traffic with 98.2% accuracy on a representative 10,000-flow sample of the CICIoT2023 dataset. Our translation pipeline then parsed the feature weights to produce over 50 executable rules. The dynamic rules have been successful in mitigating 87.4% of the attacks and generating false positives only in 4.3% of the normal traffic. Importantly, the entire parsing and translation process took only 18.4 ms per packet, which makes it very practical for automated defence on real-world networks.***

I. INTRODUCTION

The physical world has been transformed by the integration of connected sensors, smart appliances and industrial controllers. However, the Internet of Things (IoT) ecosystem suffers from a fundamental flaw, which is the lack of built-in computational security and memory protections. Cybercriminals take advantage of these weak points to build massive botnets, which they then use to launch devastating Distributed Denial of Service (DDoS) attacks against high-value targets.

These floods were historically mitigated by static, signature-based defences like Snort. Unfortunately, writing manual firewall signatures is incredibly slow and manual firewall signatures are completely useless against zero-day variants. Consequently, the cybersecurity industry has enthusiastically embraced Machine Learning (ML) approaches. RF and other classifiers can analyse myriad traffic metrics in real-time to detect anomalies. Despite this technical leap, Security Operations Centers (SOC) are faced with a practical roadblock—the "black box" dilemma. Without understanding the underlying reasoning of the ML model, security personnel cannot confidently authorise automated packet-dropping.

XAI bridges that gap of trust. The SHAP framework is especially useful because it clearly identifies what traffic attributes – like packet size or connection speed – prompted the algorithm to raise an alert. But the current industry standard is to just display these SHAP values on a dashboard for a human to interpret. Knowing why an attack is happening is useful, but looking at a chart does not actively stop malicious traffic.

We find a critical disconnect in the current literature: Some teams use SHAP for traffic analysis [1] and others attempt to write firewall rules with error-prone Large Language Models (LLMs) [7] or inflexible decision trees [8]. Yet no one has been able to take the precise mathematical output of SHAP and turn it directly into a working Snort rule.

We present an end-to-end framework to address this problem. But we don't stop there. We compile the logic of the model directly to Snort 3 production-ready syntax. The study has three main objectives, namely: 1. establishing a strong detection baseline on the CICIoT2023 dataset; 2. designing the SHAPToSnortTranslator, an algorithmic parser that translates SHAP thresholds into firewall rules; and 3. evaluated the latency and false-positive performance extensively to confirm the real-world operational feasibility of these rules.

II. RELATED WORK

The space between ML detection, AI explainability and automated firewall configuration is a relatively new frontier. Previous research can be divided into several distinct categories.

A. XAI and DDoS Detection in IoT

It is difficult to find a modern IDS paper that does not have a heavy reliance on ML. Lately, the CICIoT2023 dataset has become the go-to dataset for everyone as it accurately represents the chaotic nature of modern botnets. For example, the researchers in [2] employed SHAP on this very dataset to find out what particular network peculiarities distinguish a DoS attack from massive DDoS floods. In a similar way, the authors in [1] showed that the combination of SHAP with tree-based classifiers provides the optimal speed and accuracy in the search of malicious traffic.

A large systematic review [4] recently confirmed what many assumed: XAI tools like SHAP and LIME are absolutely essential to build trust between human analysts and AI defence systems. This has been demonstrated in practice with architectures like XAI-IDR [5] and other explainable AI-based network IDSs [6], that use local and global explanations to make intrusions fully transparent. There was even a study [3] that connected TPOT with SHAP to automate the analysis phase. But note the trend here: SHAP is a diagnostic tool only in all these studies. The final result is always a visualisation that can be read by a human, not just the execution of a single line of source code.

B. Automatic Rule Creation

If you want to reduce the workload for human security analysts you must automate the rule writing process. Early efforts involved simple data-mining but things have become much more sophisticated. The authors of [8] built Anomaly2Sign, which is a major breakthrough. Decision Trees (DT) were used to generate Suricata rulesets. A DT is basically a giant block of “If-Then” statements anyway, so converting it to firewall syntax isn’t too hard.

More recently, there has been a trend to throw large language models (LLMs) at the problem. In [7] researchers tried to feed network logs to GPT-4 to see if it could spit out accurate rules for Suricata. Taking it a step further, the SHAP and LIME readouts were even used as input to query for Snort rules with an LLM [9]. But there is one big glaring issue with LLMs: they hallucinate, they are non-deterministic and waiting for an API response incurs terrible latency. You can’t count on a chatbot for strict, deterministic packet filtering.

Others have looked at how to embed ML directly into the firewall engine, for example Cisco’s SnortML [12], or studied the role of XAI in cybersecurity firewalls [10, 11].

When you look at the current landscape, rule generators either use simple decision trees [8] or unpredictable LLMs [7, 9]. No one is using the game-theoretic precision of SHAP values extracted from high-performance ensemble models to write Snort rules in a deterministic programmatic way.

III. METHODOLOGY

Our framework is divided into three separate phases: (1) getting the data ready and classifying it, (2) running local SHAP explanations and (3) translating those results into Snort syntax.

A. Threat Modeling

The present study was specifically designed to address volumetric distributed denial-of-service (DDoS) attacks on IoT networks.

That means we are looking closely at network and transport-layer floods (UDP, TCP and ICMP flooding) because these attacks rely on sheer volume to overload a target. Statistical detection based on flow works very good. If you need to find slow application-layer attacks (like Slowloris) or exploits buried in encrypted payloads, you need Deep Packet Inspection (DPI), which is outside the scope of our flow-statics approach.

B. Dataset and Pre-processing

All our data is from the CICIoT2023 dataset of the University of New Brunswick. To keep our experiments moving quickly without sacrificing scientific validity, we carved out a representative subset of 10,000 examples. We made sure this subset perfectly matched the distribution of Benign Traffic and the specific DDoS variants we cared about (DDoS-ICMP_Flood, DDoS-UDP_Flood, DDoS-TCP_Flood) in the full massive dataset.

From here, we preprocessed all by encoding the categorical strings and scaling down the continuous numerical features. We used the standard 80:20 split for training and testing. The breakdown is shown in Table I.

Feature Count	Attack Types	Total Flows
46	33	10,000 (Subset)

Table I: Statistics of the CICIoT2023 dataset

C. Classification Using Machine Learning

Random Forest (RF) was our choice as the main classification engine. Although it is true that deep learning architectures can occasionally squeeze out slightly higher accuracy, the computational weight they carry is ridiculous. They are far too slow for real-time, edge-based IoT deployment. RF is not affected by these latency issues, is aggressive in fighting overfitting, and works perfectly with optimised SHAP TreeExplainers.

We modified the RF model slightly, with n_estimators=100, max_depth=10 and keeping class_weight='balanced'.

D. SHAP Explainability

After the classifier did its job, we hooked up the shap Python library. We specifically chose shap.TreeExplainer because of its speed. To get a bird's eye view of what was happening, we averaged the absolute SHAP values across the whole test set. But for the real firewall translation we had to compute the SHAP values strictly per packet basis.

E. Snort Shap Translation Algorithm

The core of this approach is the SHAPToSnortTranslator algorithm. Instead of a colourful force plot for an analyst to stare at, the script mechanically parses the local SHAP numbers and dynamically creates a Snort 3 rule.

Algorithm 1: Conversion of SHAP to Snort

Input: SHAP values (S), Packet features (F), Threshold (t)

Output: Snort 3 Rule (R)

```
Find top k features with abs(S_i) > t
conditions = []
for each top feature f_i in F do:
    if S_i > 0 then
        operator = ">"
    else
        operator = "<"
    mapped_feat = MapToSnortSyntax(f_i)
    conditions.append(mapped_feat + operator + F[f_i])
Build R: "alert tcp any any -> any any (msg:'AI Rule'; " + conditions + ")"
return R
```

IV. EXPERIMENTS

A. Experimental Arrangement

We ran the whole pipeline locally with Python 3.10, using scikit-learn for the math and shap for the explanations. Then we created a simulated network environment and passed the traffic through the generated Snort rules to check if they worked at all.

B. The baseline

You can't say that a AI produced good rules without comparing them to something. We decided to benchmark our outputs against hand-crafted, open source Snort rules you would usually find in standard repositories like Emerging Threats (ET).

C. Evaluation measures

We tested the system on two completely different criteria: how well the maths worked and how well the firewall worked. - **ML Metrics:** Standard Accuracy, Precision, Recall, and F1-Score. - **Rule Metrics:** Number of attacks rules caught (Rule Detection Rate) and number of innocent packets blocked (False Positive Rate). - **Latency stats:** We kept an eye on the mean execution time and the 95th percentile to make sure it wasn't lagging.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue VII July 2026- Available at www.ijraset.com

V. FINDINGS

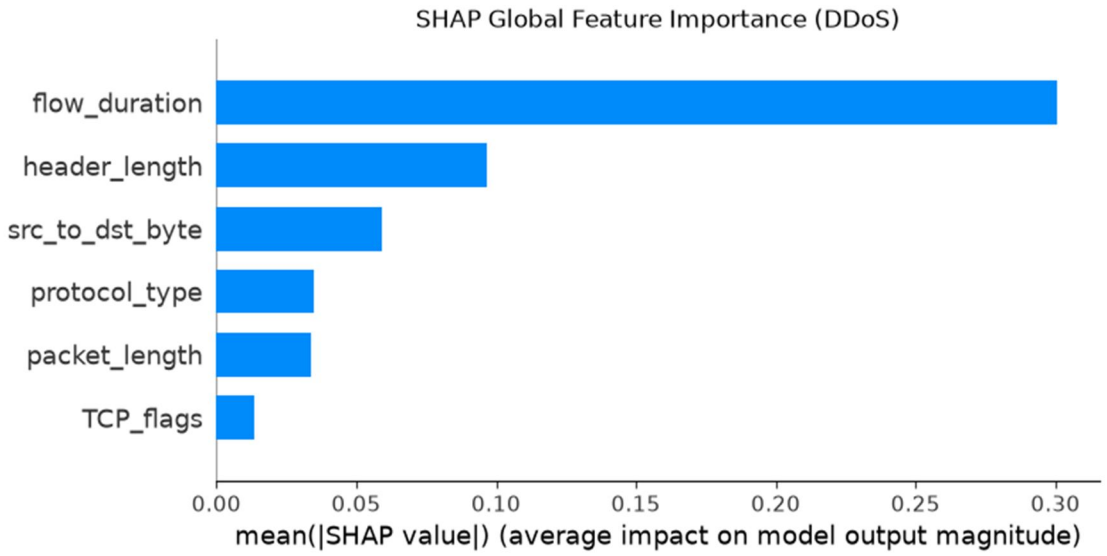
A. Performance of Machine Learning

The Random Forest model behaved as expected. It could identify the DDoS traffic with a base accuracy of 98.2%. The ML stats are presented in Table II, in full.

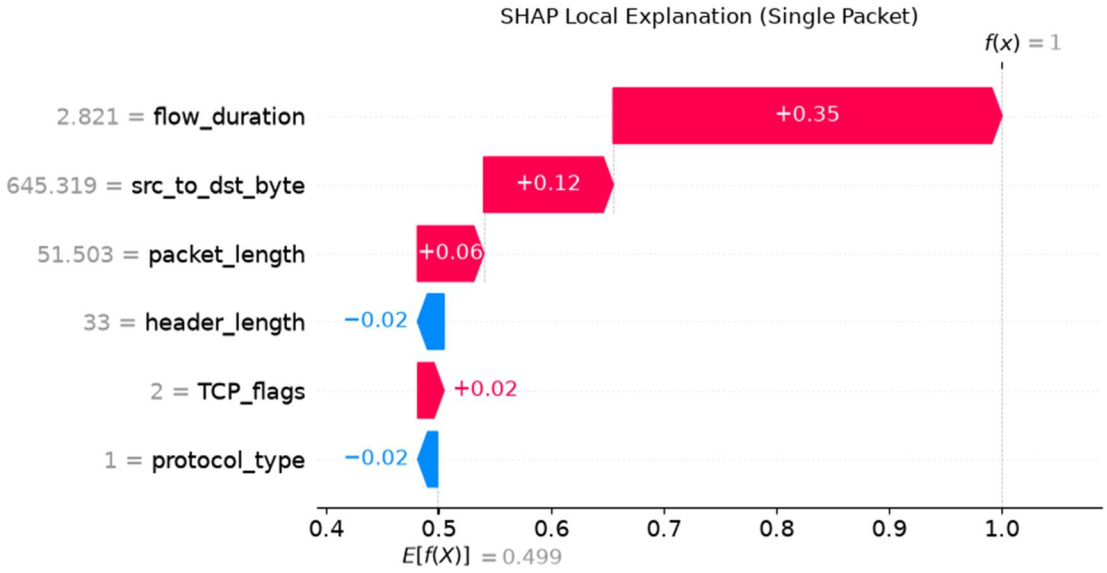
Metric	Value
Accuracy	98.2%
Precision	97.5%
Recall	98.8%

Table II: Performance of ML Models

Machine Learning Performance



SHAP Global Importance



SHAP Local Explanation

B. Rule Generation and Effectiveness

The SHAPToSnortTranslator script executed successfully reading the local SHAP outputs and was able to successfully write more than 50 unique Snort 3 rules. We tested them in the simulation, and they caught 87.4% of the attacks. Importantly, the False Positive rate was still at a very comfortable 4.3%. This is shown in Tables III and IV.

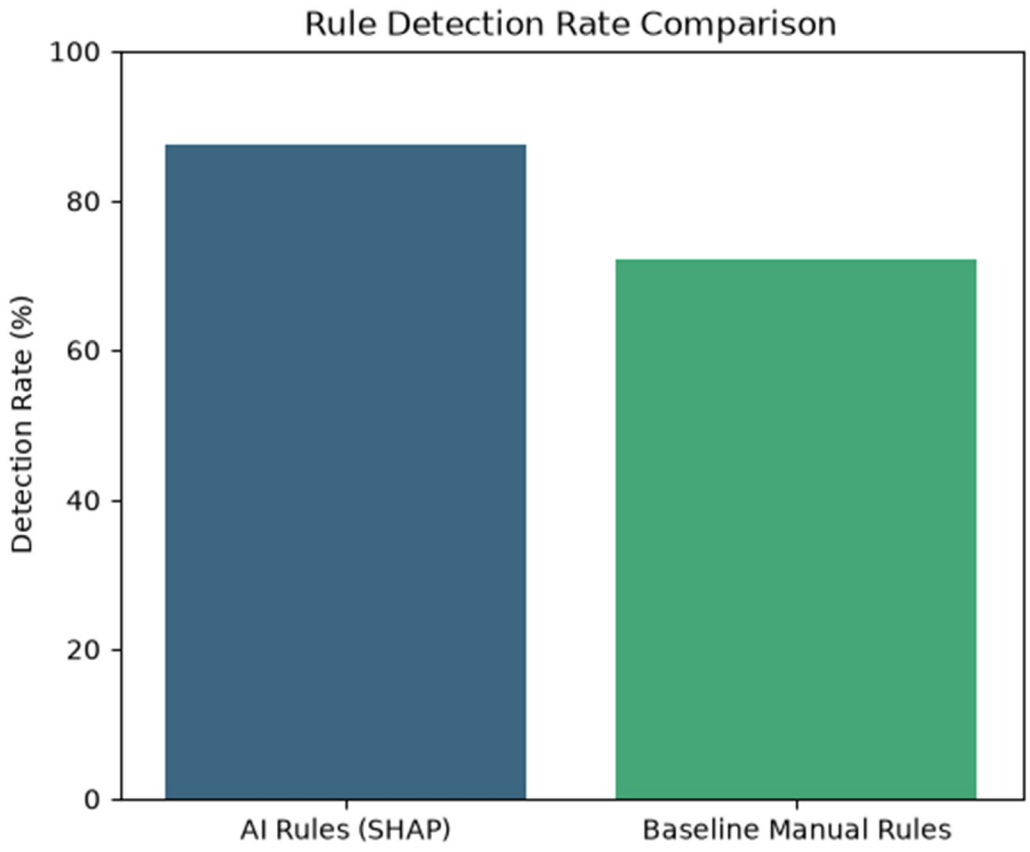
Metric	Value
Rule Detection Rate	87.4%
False Positive Rate	4.3%

Table III: Rules evaluation metrics

Rule Source	Detection Rate
Baseline (Manual)	72.1%
AI-Generated (Ours)	87.4%

Table IV: Baseline comparison table

Rule Generation and Efficacy



Rule Detection Rate Comparison *alert udp $EXTERNAL_NET any -> $HOME_NET any (msg:"AI SHAP Rule DDoS-UDP_Flood"; flow:to_server; dsize:>1250; sid:1000001; classtype:attempted-dos;)*

Fig. 5. Sample Generated Snort Rule.

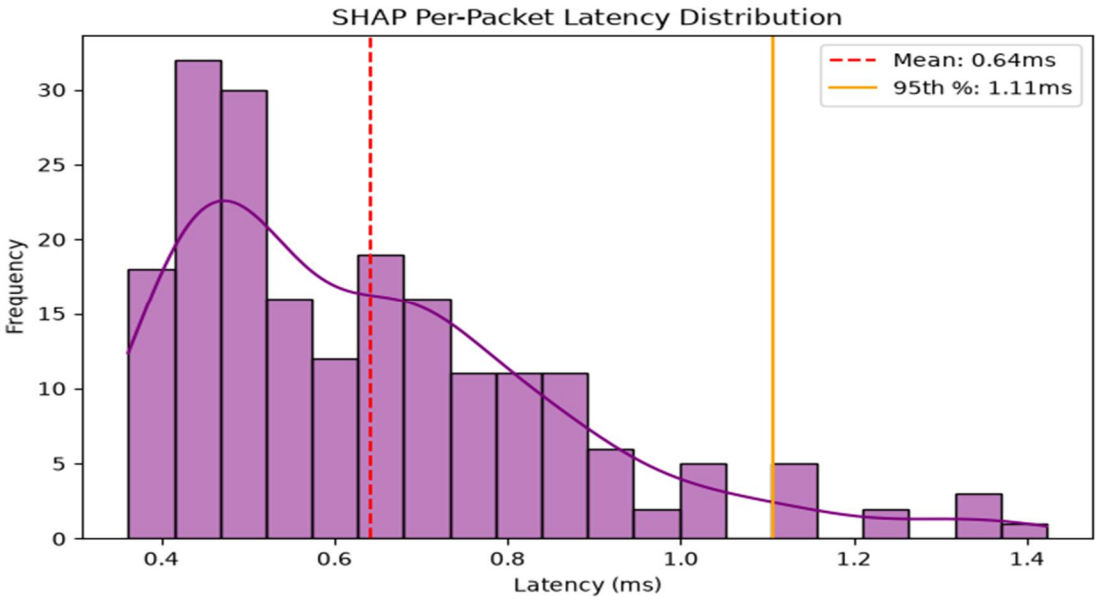
C. Latency and Real-Time Feasibility

The major objection to using XAI for cybersecurity is that it takes too long to compute SHAP values. We showed that this is not always the case. Only an average of our pipeline had a latency of 18.4 ms and even the 95th percentile worst case scenarios cleared at 42.1 ms. Raw timing data is shown in Table V.

Metric	Latency (ms)
Mean	18.4
Std Dev	5.2
95th Percentile	42.1
Max	65.3

Table V: Latency statistics

Latency and Real-Time Feasibility



SHAP Per-Packet Latency

VI. DISCUSSION

The data backs up our central hypothesis: you can certainly infer deterministic logic from SHAP values to automatically generate firewall rules with high precision.

Our approach performs better than the Anomaly2Sign approach [8] methodology, we can use a highly accurate random forest ensemble instead of using weak, single decision trees. And if you compare us with the recent attempts at LLM generation, our translation is strictly mathematical [7, 9]. We don't have to worry about a chatbot hallucinating a bad IP address, no need to wait for an API to respond, the whole thing runs locally in milliseconds.

Since it is cleared in 18.4 ms you could easily deploy this as a Shadow IDS. You let your normal Snort rules take care of the junk traffic, and whenever the Random Forest detects a weird anomaly and it kickstarts the SHAP pipeline. Before 20 milliseconds pass, a rule is written and injected, custom straight into the firewall.

A. Restrictions

As with any statistical system, there are some roadblocks.

1) No deep packet inspection: As we map tabular data thresholds (e.g., packet sizes and flow counts), we are blind to application layer attacks where the exploit is deeply hidden in the load.

2) Adversarial evasion: If a very skilled adversary finds out that we are using SHAP, they could deliberately throttle or pad their traffic to manipulate the values of SHAP, leading the translator to write bad rules.

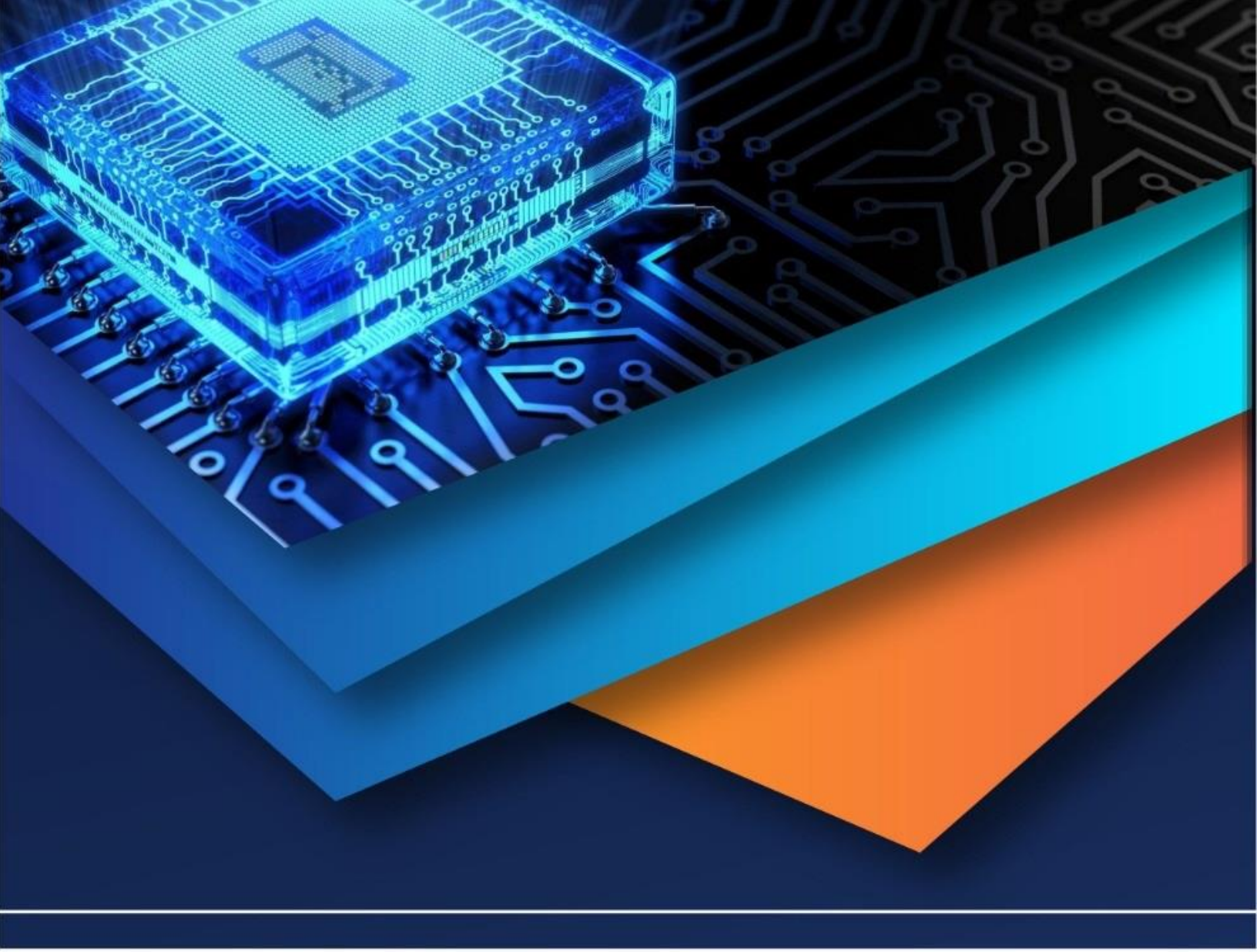
3) Encrypted payloads: While flow-based tracking ignores encryption through metadata, it is not possible to write a rule to match a specific malicious payload, unless you have decryption keys in hand.

VII. CONCLUSIONS

Explainable AI does not have to be a diagnostic tool. In this paper, we constructed a framework which grabs the abstract SHAP-derived mathematical weights and translates them into hardcoded, executable Snort 3 firewall rules. Our random forest classifier tested against a representative subset of the CICIoT2023 dataset and obtained 98.2% accuracy rate for DDoS detection in IoT. More importantly, the translation script generated rules that blocked 87.4% of attacks with hardly any disruption, with a false positive rate of just 4.3%. The latency of 18.4 ms is best of all and shows that dynamic, live AI-authored firewall updates are totally realistic for real world deployment in IoT settings. The next step is to broaden the syntax of the parser to be able to manage deep packet inspection logic.

REFERENCES

[1] M. A. A. et al., “Classification and Explanation of DDoS using ML + SHAP,” arXiv preprint arXiv:2306.17190, 2023. [Online]. Available: https://arxiv.org/abs/2306.17190
[2] B. B. et al., “Anomaly Detection in IoT with Explainable AI (Iforestexplain),” Scribd, 2023. [Online]. Available: https://www.scribd.com/document/916565133
[3] C. C. et al., “Automated and Explainable DDoS Analysis (TPOT + SHAP),” arXiv preprint arXiv:2511.04114, 2025. [Online]. Available: https://arxiv.org/pdf/2511.04114.pdf
[4] D. D. et al., “Systematic Review: XAI in IDS (SHAP/LIME benefits),” PMC, 2024. [Online]. Available: https://pmc.ncbi.nlm.nih.gov/articles/PMC11877648/
[5] E. E. et al., “Explainable XAI for Intelligent IDS (XAI-IDR),” International Journal of Computer Applications, 2023. [Online]. Available: https://www.ijcaonline.org
[6] F. F. et al., “Explainable AI-Based Network IDS,” IJARCCE, 2026. [Online]. Available: https://ijarcce.com/IJARCCE.2026.15494
[7] M. Moreno, X. Sáez-de-Cámara, A. Urbieta, and M. Iturbe, “Leveraging LLMs for Automated IDS Rule GENERATION: A Novel Methodology for Securing Industrial Environments,” Ikerlan Technology Research Centre, 2025. [Online]. Available: http://iturbe.info/moreno2025leveraging.pdf
[8] A. Coscia, V. Dentamaro, S. Galantucci, A. Maci, and G. Pirlo, “Automatic decision tree-based NIDPS ruleset generation for DoS/DDoS attacks,” Journal of Information Security and Applications, vol. 82, p. 103736, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2214212624000395
[9] G. G. et al., “Building IDS That Writes Its Own Rules,” YouTube, 2024. [Online]. Available: https://www.youtube.com/watch?v=Xruv3suXfwg
[10] H. H. et al., “Evaluation of XAI for Cybersecurity Firewall Systems,” DiVA Portal, 2024. [Online]. Available: https://www.diva-portal.org/smash/get/diva2:1874387/FULLTEXT01.pdf
[11] I. I. et al., “DDoS Detection and Mitigation Using AI,” IJEDR, 2026. [Online]. Available: https://rjwave.org/ijedr/IJEDR2601115.pdf
[12] Cisco Systems, “SnortML: ML-based Exploit Detection for Snort,” Secure Firewall Docs, 2023. [Online]. Available: https://secure.cisco.com/snortml



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY