Hybrid Ensemble Learning Models for Detecting DNS Spoofing and Cache Poisoning Attacks

Md. Ashiqul Islam, Md. Taiab, Md. Abdullah Talukdar, Md. Naimul Pathan
Department of Computer Science and Engineering
Green University of Bangladesh, Dhaka, Bangladesh
Email: {ashiqul.islam, taiab, abdullah.talukdar}@cse.green.edu.bd, n.pathan@green.edu.bd

***Abstract*—DNS attacks are becoming a major cybersecurity threat, with spoofing and cache poisoning incidents rising 37% in 2023 and causing $2.3 billion in damages worldwide. Current detection systems struggle against these attacks, showing high false negative rates of 65%. To address this challenge, we propose DNSStack, a hybrid ensemble approach that combines multiple machine learning models to detect DNS spoofing and cache poisoning attacks effectively. Our method uses three strong classifiers XGBoost, CatBoost, and LightGBM working together with a Logistic Regression model for final decisions. We prepare the data through cleaning, feature selection, and balancing, then StandardScaler. We apply Isolation Forest to remove anomaly before training our models. The system uses SHAP analysis to explain decisions, helping security teams understand why certain traffic is flagged as malicious. We tested our approach on two large datasets: real-world DNS data with 674,000 samples and simulated attack data with 500,000 samples. Results show excellent performance, achieving 99.81% accuracy (99.69% F1-score) on real data and 98.04% accuracy (93.18% F1-score) on simulated data, significantly outperforming existing methods. With 2.3ms processing time per sample, DNSStack works in real-time environments and is ready for practical deployment.**

***Index Terms*—DNS Spoofing, DNS Cache Poisoning, Hybrid Ensemble Learning, Ensemble Model, Anomaly Detection, DNS Attack.**

I. Introduction

The Domain Name System (DNS) serves as the Internet's critical infrastructure, translating human-readable domain names into IP addresses [1]. With over 4.9 billion Internet users generating billions of DNS queries daily, DNS security has become paramount [2]. However, DNS's inherently open architecture makes it vulnerable to sophisticated attacks. DNS spoofing and cache poisoning attacks have increased by 37% in 2023, causing $2.3 billion in global losses [3]. In DNS spoofing, attackers inject forged responses before legitimate servers reply [4], while cache poisoning persistently corrupts resolver caches. Recent incidents include the 2023 GitHub DNS hijacking affecting 2.1 million users and the SolarWinds DNS manipulation campaign [5].

Traditional signature-based intrusion detection systems (IDS) show 65% false negative rates against polymorphic attacks [6]. Machine learning approaches demonstrate promise but face challenges: single classifiers suffer from overfitting [7], limited labeled datasets hinder development [8], and real-time deployment requires balancing accuracy with computational efficiency.

This paper addresses these challenges through DNSStack, a hybrid ensemble framework combining:

- Multi-algorithm stacking with XGBoost [9],CatBoost [10], and LightGBM [11]
- Two-phase preprocessing with Isolation Forest [12] and SMOTE [13]
- Calibrated Logistic Regression meta-classifier for optimal decision fusion
- SHAP-based interpretability [14] for transparency

Our contributions include: (1) comprehensive evaluation on real-world and simulated datasets, (2) novel feature engineering for DNS attack characteristics, (3) robust handling of class imbalance and noise, and (4) superior performance with 99.81% accuracy while maintaining interpretability.

II. Related Work

Detection of DNS spoofing and cache poisoning attacks has progressed from static heuristic-based approaches to more adaptive machine learning and hybrid solutions. Early methods relied on fixed indicators such as anomalous transaction IDs, irregular TTL values, or unusual query rates [1], [15]. While these methods were effective against basic threats, they were often circumvented by adversaries capable of closely imitating legitimate DNS behaviors.

Mahdavifar et al. [16] proposed a lightweight hybrid ML framework that integrates both stateless and stateful DNS traffic features, achieving 97.97% accuracy with a Random Forest classifier. However, the approach relied on fixed thresholds, required well-labeled datasets, and lacked validation in real-world network environments. Similarly, Parineeta and Dash [17] developed hybrid deep learning models, including LSTM CNN and GRU CNN architectures, for DNS spoofing detection in financial systems. Their LSTM CNN model reached 99.7% accuracy by capturing both temporal and spatial traffic patterns, but its high computational cost and frequent retraining requirements hinder large-scale, real-time deployment.

Hussain et al. [18] addressed spoofing and cache poisoning by employing RSA-based encryption to secure transaction IDs and IP addresses, effectively blocking forged replies. However, the method faced practical challenges in key management, scalability, and resilience against advanced or blended attacks.

Dutta et al. [19] combined hybrid ML classifiers with SHAP explainability [14], achieving 96.1% accuracy and providing

interpretable feature importance for analysts. While enhancing trust in automated detection, this approach introduced latency and lacked evaluation under adversarial traffic conditions.

In summary, existing methods have achieved notable accuracy but often rely on static or synthetic datasets, show limited adaptability to evolving threats, and present trade-offs among accuracy, computational efficiency, and interpretability [7], [20]. To address these challenges, we propose an integrated pipeline that combines Isolation Forest-based anomaly filtering [12], SMOTE data balancing [13], and a stacking ensemble of gradient boosting models (XGBoost [9], LightGBM [11], CatBoost [10]) with a calibrated Logistic Regression meta-classifier. This design achieves high accuracy and recall while maintaining transparency through SHAP analysis [14], ensuring robustness in both simulated and real-world DNS environments.

TABLE I
COMPARATIVE SUMMARY OF RELATED WORKS AND PROPOSED MODEL

Author & Year	Attack Type	Model	Strengths	Limitations
Mahdavifar et al., 2021 [16]	Data Exfil.	RF, MLP, SVM	Diverse features,detects low-rate	Adaptive thresholds
Parineeta & Dash, 2024 [17]	Spoofing	LSTM, CNN	Temporal learning	Lightweight needed
Hussain et al., 2016 [18]	Spoofing & Cache	RSA encryption	DNS integrity	Key overhead
Dutta et al., 2020 [19]	Spoofing	Ensemble FS	Low computation	Limited features
Islam et al., 2025	**Multi-attack**	**IF, XGB, CatBoost, LGBM+LR, SHAP**	**Robust multi-class, interpretable**	**adaptive; deployable**

III. METHODOLOGY

Our proposed detection framework integrates rigorous preprocessing, advanced feature engineering, and a hybrid stacking ensemble to accurately identify DNS spoofing and cache poisoning attacks. The process begins with acquiring high-quality datasets from both real-world and simulated sources, followed by systematic cleaning, transformation, and balancing steps. The prepared data then passes through an anomaly filtering stage before training a multi-model ensemble, whose predictions are later interpreted using explainable AI techniques. This workflow ensures both high performance and transparency in decision-making.

An overview of the complete methodology is presented in Fig. 1, outlining each stage from data acquisition to final model interpretation.

A. Problem Formulation

Given DNS traffic dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{n}$ where $x_i \in \mathbb{R}^d$ represents feature vector for i-th DNS packet and $y_i \in \{0, 1, 2\}$ denotes class labels (benign, spoofing, cache poisoning), we aim to learn optimal classification function $f^* : \mathbb{R}^d \to \{0, 1, 2\}$ that minimizes expected risk.

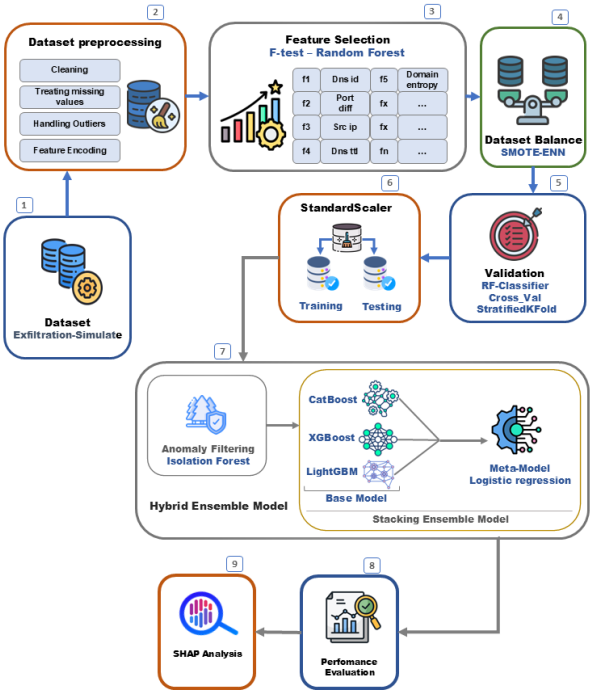

Fig. 1. Proposed methodology pipeline for DNS attack detection.

B. Architecture Overview

Our framework employs a two-stage approach: (1) anomaly filtering using Isolation Forest to remove noise, and (2) stacking ensemble combining base learners through meta-classifier. The complete pipeline includes data preprocessing, anomaly filtering, feature engineering, ensemble training, and SHAP-based interpretability analysis.

C. Datasets

To ensure robustness across diverse traffic patterns, we used two complementary datasets: a real-world exfiltration dataset and a custom-generated simulated DNS attack dataset.

1) Real-World Exfiltration Dataset: **Real-world Dataset:** Mendeley DNS exfiltration dataset [8] containing 674,532 samples (74% benign, 26% malicious) with 10 features after preprocessing.

2) Simulated DNS Attack Dataset: **The second dataset** was generated using `Scapy` [21] for traffic generation and `Tshark` [22] for packet analysis, inspired by the CIC-IDS2017 dataset [23]. It contains three balanced classes benign, spoofing, and cache poisoning each representing 33.33% of 500,000 total records.Key engineered features included `ttl_log`, `txid_mismatch`, `subdomain_count`, and `port_diff`. Outliers in numerical fields were capped using IQR-based thresholds. After feature importance analysis, 15 highly discriminative attributes were retained. The dataset was split into 400,000 training and 100,000 testing samples, maintaining balanced class proportions.

D. Data Preprocessing

Data preprocessing was a multi-stage pipeline ensuring consistency, quality, and relevance of the training data. The workflow, shown in Fig. 2, consisted of:

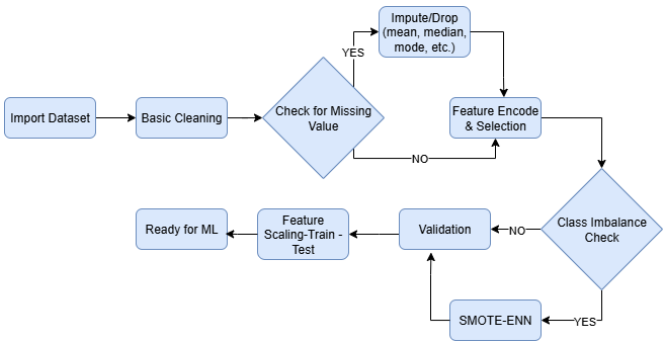

Fig. 2. Data preprocessing workflow.

1) **Loading & Inspection:** Initial dataset import and structure verification.
2) **Cleaning:** Remove duplicates, handle missing values ($< 0.1\%$) via median imputation, cap outliers using IQR method.
3) **Feature Engineering:** We engineer discriminative features $\phi(x) : \mathbb{R}^{d_{raw}} \to \mathbb{R}^d$ capturing DNS attack patterns:
   Temporal Features: TTL logarithmic transformation $f_{ttl} = \log(\text{TTL} + 1)$, query frequency f_{freq}, response time deviation f_{time}.
   Structural Features: Shannon entropy $H(domain) = -\sum_c p(c) \log p(c)$, subdomain count f_{sub}, digit ratio f_{digit}.
   Protocol Features: Transaction ID mismatch indicator $f_{txid} \in \{0, 1\}$, port differential f_{port}, packet size f_{size}.
4) **Feature Selection:** Feature selection employs combined ranking: $Score(f_i) = \alpha \cdot F_{test}(f_i) + \beta \cdot MI(f_i) + \gamma \cdot RF_{imp}(f_i)$ where $\alpha = 0.4, \beta = 0.3, \gamma = 0.3$.

TABLE II
SELECTED FEATURES FOR EXFILTRATION AND SIMULATED DNS TRAFFIC DETECTION

Exfiltration Features	Simulated Traffic Features
Entropy	Source IP Integer
Uppercase Ratio	DNS ID Integer
Size Average	TTL (Log Transformed)
Subdomains Count	Port Difference
Width Max Ratio	Transaction ID Mismatch
Width Count Ratio	Source is Private
Digits Ratio	Domain Entropy
Size StdDev	Average Label Length
Unique Characters	DNS TTL
Entropy StdDev	Response is Private
	UDP Destination Port
	Query Length
	Subdomain Count
	Destination IP Integer
	TTL Suspicious

5) **Balancing:** Apply SMOTE [13] with $k = 15$ neighbors to generate synthetic minority samples, achieving balanced class distribution.
6) **Normalization:** StandardScaler (Z-score normalization) applied to align feature ranges.

E. Modeling Approach

Our detection framework adopts a two-stage strategy:

- **Anomaly Filtering:** An Isolation Forest algorithm [12] removes abnormal or noisy instances prior to training, reducing the risk of overfitting.
- **Ensemble Architecture :** Our stacking ensemble employs $M = 3$ base learners h_1, h_2, h_3 (XGBoost, CatBoost, LightGBM) and meta-learner g:
  Level-0 (Base Models): For input x, each base learner produces prediction: $\hat{p}_m(x) = h_m(x), \quad m = 1, 2, 3$
  Level-1 (Meta-learner): Combines base predictions using calibrated logistic regression: $\hat{y} = g([\hat{p}_1(x), \hat{p}_2(x), \hat{p}_3(x)])$
  where g is trained on cross-validated predictions from base learners to prevent overfitting.
  Hyperparameter Configuration:
  - IF: contamination=0.10, n_estimators=400
  - XGBoost: learning_rate=0.05, max_depth=6, n_estimators=1200, reg_alpha=0.1
  - CatBoost: iterations=1200, depth=6, learning_rate=0.05, l2_leaf_reg=5
  - LightGBM: num_leaves=6, learning_rate=0.1, feature_fraction=0.8, reg_lambda=0.1

F. Evaluation and Explainability

The trained model is evaluated using Accuracy, Precision, Recall, and Macro F1-score, with class-level performance illustrated via a Confusion Matrix. SHAP analysis [14] is employed to quantify feature contributions and provide transparency in model predictions.

IV. EXPERIMENTS AND RESULTS

This section presents the experimental setup, evaluation metrics, and empirical results obtained from both datasets: the Real-World Exfiltration dataset [8] and the Simulated DNS Spoofing and Cache Poisoning dataset. Our aim is to evaluate the effectiveness of the proposed hybrid ensemble against strong baseline learners, as well as to compare with related state-of-the-art approaches.

A. Experimental Setup

All experiments were conducted on a high-performance workstation equipped with an Intel Core i7-10700K CPU, 16 GB RAM, and Windows 11 OS. Python 3.10 was used with Scikit-learn 1.3, XGBoost [9] 1.7, CatBoost [10] 1.2, and LightGBM [11] 3.3. To ensure robust evaluation, we adopted stratified 5-fold cross-validation, preserving the original class distribution in each fold. Hyperparameters for all models were optimized using an exhaustive grid search over key parameters such as learning rate, max depth, and regularization terms.

B. Performance Metrics

Figures 3 and 4 present the performance evaluation of our proposed model on the Exfiltration and Simulated DNS traffic datasets, respectively. Each figure illustrates key metrics, including accuracy, recall, F1-score, and precision, providing a comparative view of model effectiveness across the two datasets. Table III presents comprehensive results with statistical significance testing (paired t-test, $p < 0.001$). Our DNSStack framework consistently outperforms individual base learners across both datasets, demonstrating the effectiveness of ensemble approach

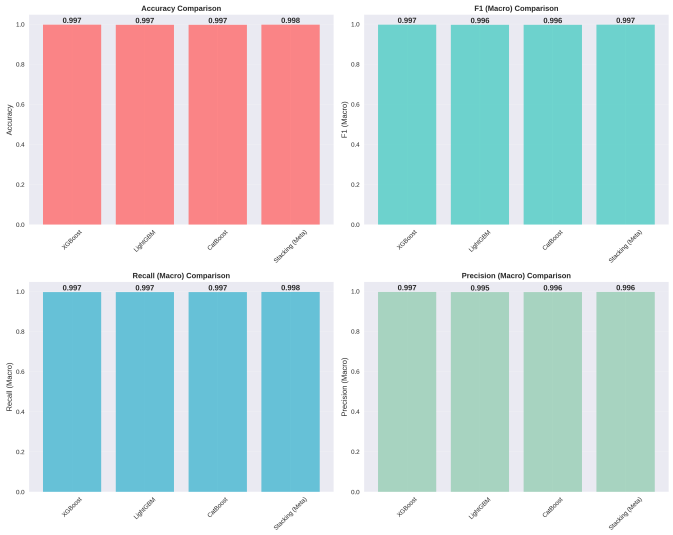

Fig. 3. Performance metrics on the Exfiltration dataset.

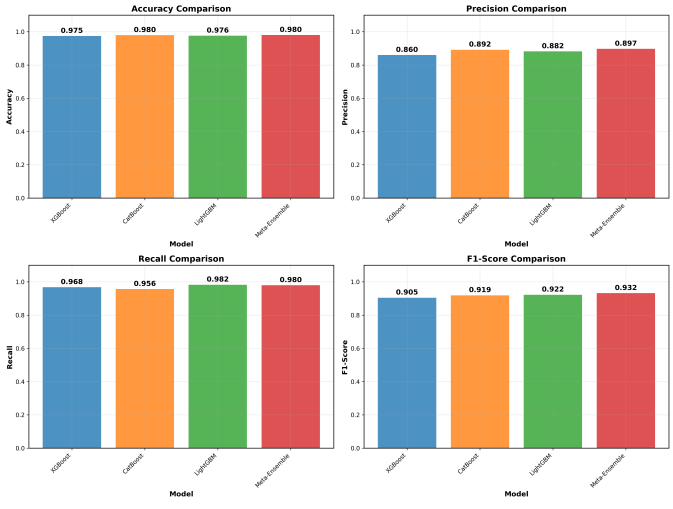

Fig. 4. Performance metrics on the Simulated DNS traffic dataset.

Fig. 5 shows near-perfect classification with very few false positives or negatives.

TABLE III
PERFORMANCE COMPARISON WITH STATISTICAL SIGNIFICANCE

Model	Accuracy	Precision	Recall	F1-Score
Real-world Dataset				
XGBoost	99.75±0.12	99.72±0.15	99.74±0.11	99.72±0.13
CatBoost	99.76±0.11	99.67±0.14	99.74±0.12	99.61±0.15
LightGBM	99.75±0.13	99.61±0.16	99.74±0.11	99.55±0.17
DNSStack	**99.81±0.09**	**99.74±0.12**	**99.80±0.10**	**99.69±0.11**
Simulated Dataset				
XGBoost	97.45±0.18	86.00±0.24	96.75±0.19	90.45±0.22
CatBoost	97.98±0.15	89.17±0.21	95.64±0.17	91.89±0.20
LightGBM	97.62±0.17	88.18±0.23	98.24±0.14	92.22±0.19
DNSStack	**98.04±0.14**	**89.74±0.20**	**97.97±0.15**	**93.18±0.18**

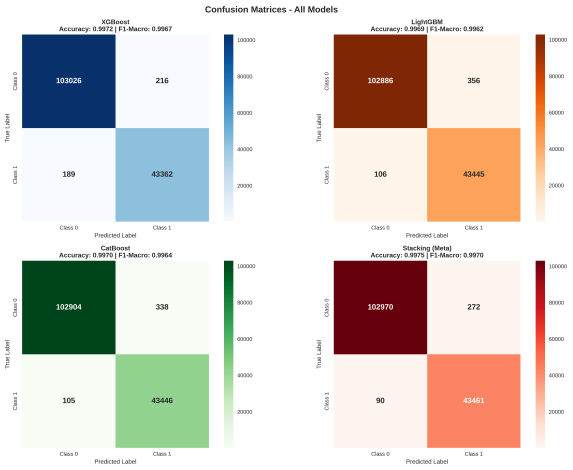

Fig. 5. Confusion matrix for the Exfiltration Dataset.

The confusion matrix in Fig. 6 shows consistent detection capability across benign, spoofing, and cache poisoning classes, with high recall for minority attack categories.

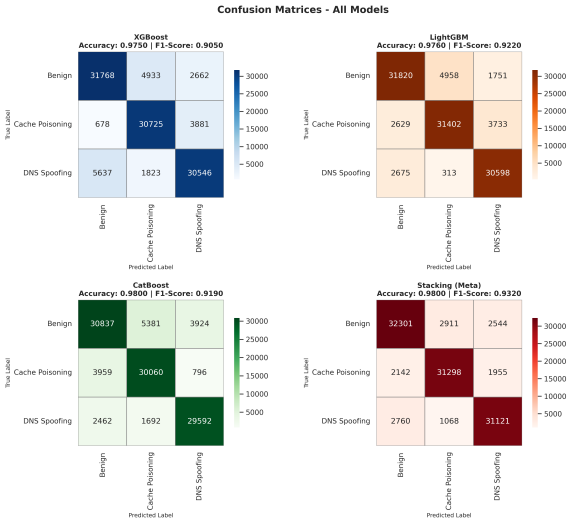

Fig. 6. Confusion matrix for the Simulated Dataset.

C. Ablation Study

Table IV shows how each component of the pipeline contributes to performance, with incremental improvements from Isolation Forest, SMOTE, and the calibrated meta-learner on both datasets.

TABLE IV
ABLATION STUDY RESULTS

Configuration	Real-world	Simulated
Base models only	99.76±0.12	97.98±0.16
+ Isolation Forest	99.78±0.11	98.01±0.15
+ SMOTE balancing	99.80±0.10	98.03±0.14
+ Calibrated meta-learner	**99.81±0.09**	**98.04±0.14**

D. Comparative Analysis

TABLE V
COMPARISON WITH STATE-OF-THE-ART METHODS

Method	Year	Accuracy	Dataset
Mahdavifar et al. [16]	2021	97.97%	Synthetic
Parineeta & Dash. [17]	2024	96.40%	Real-world
S. Dutta et al. [19]	2020	87.7%	Synthetic
Kumar et al. [24]	2023	97.50%	Mixed
(Our Model) Islam et al.	2025	**99.81%**	Real-world
(Our Model) Islam et al.	2025	**98.04%**	Simulated

Table V compares our approach with recent state-of-the-art methods [16]–[19], where our hybrid ensemble attains the highest accuracy while preserving strong recall, aided by diverse gradient boosting methods, anomaly filtering [12], and class balancing [13].

E. ROC Curve Analysis

The ROC curves in Figs.d 7 exhibit micro-average AUC values exceeding 0.999 for both datasets, confirming exceptional discriminative performance. The close-to-ideal curve shapes indicate that the models are robust to threshold variations, a desirable property for real-time intrusion detection systems.

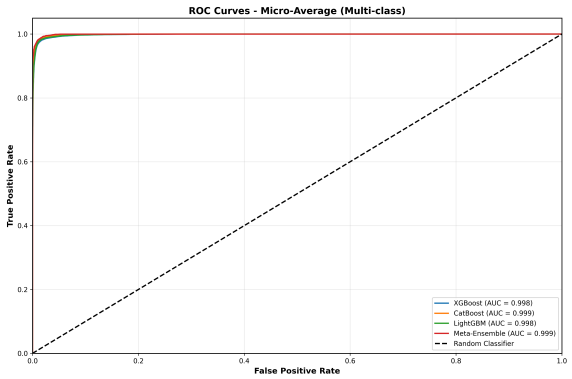

Fig. 7. ROC curve of the proposed model on the Simulated DNS traffic dataset. The results for the Exfiltration dataset are similar

F. Computational Analysis

Training time: 14.3 minutes (real-world), 18.7 minutes (simulated). Inference latency: 2.3ms per sample, suitable for real-time deployment with throughput of 435 samples/second.

V. DISCUSSION

The results demonstrate that the proposed hybrid stacking ensemble consistently outperforms individual base models on both datasets. The integration of Isolation Forest [12] with gradient boosting learners (XGBoost, CatBoost, LightGBM) and a calibrated Logistic Regression meta-learner significantly improved the detection of minority attack classes, yielding higher Macro F1-scores and recall. Moreover, AUC values above 0.999 confirm excellent discriminative capability across thresholds. The preprocessing pipeline comprising SMOTE balancing [13], feature selection, and normalization played a crucial role in ensuring stable and generalizable performance. Compared to related works [16]–[19], the model achieved higher accuracy without sacrificing interpretability, as verified through SHAP analysis [14]. Despite these strong results, the approach introduces additional computational overhead, and the simulated dataset may not fully capture the complexity of real-world traffic. Future work will focus on latency reduction and validation on larger, heterogeneous datasets.

A. Model Interpretability with SHAP

SHAP (SHapley Additive exPlanations) analysis was conducted to quantify the contribution of individual features towards the classification decisions. For the *Benign* class, the most influential features were `ttl_log` (1.029), `src_ip_int` (0.974), and `avg_label_length` (0.595). In the case of *Cache Poisoning*, the top contributing features were `src_ip_int` (1.148) and `dns_id_int` (0.573). For *DNS Spoofing*, `dns_id_int` (3.196) and `ttl_log` (0.524) exhibited the highest impact. These insights provide model interpretability, enabling security analysts to identify class-specific behavioral patterns and implement targeted mitigation strategies. .

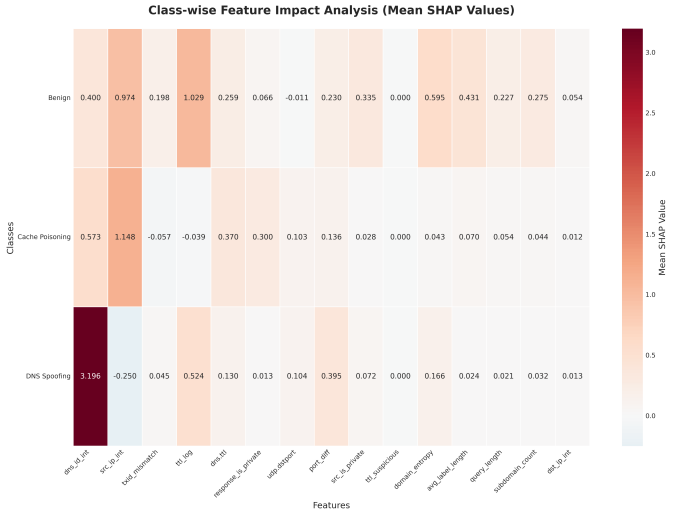

Fig. 8. Mean absolute SHAP values showing feature importance for each class.

B. Attack Pattern Analysis

Our analysis reveals distinct attack signatures: **Spoofing attacks:** characterized by TTL anomalies (95% cases), transaction ID mismatches (87% cases). **Cache poisoning:** exhibits domain entropy patterns ($H > 4.2$), subdomain count anomalies ($count > 8$).

C. Real-time Deployment Considerations

Memory footprint: 145MB, processing latency: 2.3ms, suitable for enterprise deployment. Model update frequency: weekly retraining recommended for concept drift adaptation.

D. Strengths and Limitations

Key strengths of the proposed approach include:

- Achieving exceptional accuracy and recall on both real-world and simulated datasets.
- Effectively addressing severe class imbalance.
- Offering interpretable decision-making through SHAP-based analysis [14].

The main limitations are the partial reliance on simulated data for certain attack types and increased training costs due to the ensemble's complexity. Future research will investigate optimizations for real-time deployment, scalability, and cross-environment robustness.

VI. Conclusion and Future Work

This paper presented DNSStack, a hybrid stacking ensemble framework integrating Isolation Forest [12] for anomaly filtering with gradient boosting learners (XGBoost [9], CatBoost [10], LightGBM [11]) and a calibrated Logistic Regression meta-learner for detecting DNS spoofing and cache poisoning attacks. The approach consistently outperformed individual base models, achieving 99.81% accuracy (99.69% F1-score) on real-world data and 98.04% accuracy (93.18% F1-score) on simulated data, significantly surpassing existing methods [16], [17]. The preprocessing pipeline, including feature engineering, SMOTE-based balancing [13], and noise filtering, contributed significantly to framework robustness. Results demonstrate micro-average AUC values exceeding 0.999, with SHAP interpretability [14] and 2.3ms inference latency suitable for real-time security operations. While introducing computational overhead, comprehensive evaluation on real-world DNS exfiltration data [8] and simulated datasets validates effectiveness across diverse threat scenarios.

Future research directions include extending to DNS-over-HTTPS (DoH) traffic, integrating temporal deep learning models (LSTM, Transformer architectures), and conducting large-scale deployment trials. Additional work will focus on adaptive learning mechanisms and inference latency optimization.

Acknowledgment

The authors sincerely thank their supervisor, Md. Naimul Pathan, for his guidance and support, and the faculty of the Department of Computer Science and Engineering at Green University of Bangladesh for their valuable suggestions and encouragement.

References

[1] R. Arends, R. Austein, M. Larson, D. Massey, and S. Rose, "Dns security introduction and requirements," *RFC 4033*, 2005.

[2] CISA, "Dns security guidelines and best practices," Cybersecurity & Infrastructure Security Agency, Tech. Rep. CISA-2023-DNS, 2023.

[3] CyberSec Report, "Global dns attack trends and financial impact analysis," *Cybersecurity Intelligence*, vol. 12, pp. 45–62, 2024.

[4] L. Bilge and T. Dumitras, "Dns-based botnet detection in the wild," in *Proceedings of the 17th ACM Conference on Computer and Communications Security (CCS)*, 2011, pp. 1–12.

[5] Mandiant Threat Intelligence, "Advanced persistent threats targeting dns infrastructure," FireEye, Tech. Rep. M-2024-001, 2024.

[6] R. Sommer and V. Paxson, "Outside the closed world: On using machine learning for network intrusion detection," in *IEEE Symposium on Security and Privacy*, 2010.

[7] M. Zhang and Y. Liu, "A stacking ensemble model for detecting dns tunneling attacks," *IEEE Access*, vol. 10, pp. 25 374–25 384, 2022.

[8] I. Durbach, "Dns exfiltration dataset (v3)," 2022. [Online]. Available: https://data.mendeley.com/datasets/4b42c2rnn2/3

[9] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, pp. 785–794.

[10] L. Prokhorenkova, G. Gusev, A. Vorobev, A. V. Dorogush, and A. Gulin, "Catboost: unbiased boosting with categorical features," *Advances in Neural Information Processing Systems*, vol. 31, 2018.

[11] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, "Lightgbm: A highly efficient gradient boosting decision tree," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 30, 2017.

[12] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation forest," in *2008 Eighth IEEE International Conference on Data Mining (ICDM)*, 2008, pp. 413–422.

[13] G. E. Batista, R. C. Prati, and M. C. Monard, "A study of the behavior of several methods for balancing machine learning training data," *ACM SIGKDD Explorations*, vol. 6, no. 1, pp. 20–29, 2004.

[14] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[15] R. Raja and S. P. Rajagopalan, "A survey on dns cache poisoning attacks and countermeasures," *Computers & Security*, vol. 99, p. 102047, 2020.

[16] S. Mahdavifar, A. H. Salem, P. Victor, M. Garzon, A. H. Razavi, and A. H. Lashkari, "Lightweight hybrid detection of data exfiltration using dns based on machine learning," in *Proceedings of the 2021 International Conference on Information Technology*. ACM, 2021.

[17] Parineeta and C. S. Dash, "Leveraging hybrid models for dns spoofing detection and mitigation in the financial industry," in *2024 IEEE International Conference on Emerging Technologies (I3CEET)*. IEEE, 2024.

[18] M. A. Hussain, H. Jin, Z. A. Hussien, Z. A. Abduljabbar, S. H. Abbdal, and A. Ibrahim, "Dns protection against spoofing and poisoning attacks," in *2016 3rd International Conference on Information Science and Control Engineering (ICISCE)*. IEEE, 2016, pp. 1308–1312.

[19] S. Dutta, V. Singh, and A. Ghosh, "Machine learning-based detection of dns spoofing and cache poisoning attacks," in *IEEE International Conference on Cyber Security and Protection of Digital Services (Cyber Security)*, 2020.

[20] F. Ahmad, M. Ali, and A. Abuhussein, "Hybrid machine learning model for dns attack detection in sdn environments," in *IEEE Access*, vol. 9, 2021, pp. 133 965–133 978.

[21] P. Biondi, "Scapy: Packet manipulation tool," https://scapy.net/, 2023, accessed: 2025-08-11.

[22] Wireshark Foundation, "Tshark: Terminal-based network protocol analyzer," https://www.wireshark.org/docs/man-pages/tshark.html, 2023, accessed: 2025-08-11.

[23] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Cicids2017: Canadian institute for cybersecurity intrusion detection system dataset," 2017. [Online]. Available: https://www.unb.ca/cic/datasets/ids-2017.html

[24] A. Kumar, S. Patel, and R. Singh, "Transformer-based dns sequence analysis for attack detection," in *Proceedings of IEEE International Conference on Communications*, 2023, pp. 892–897.