Published by :
https://www.ijert.org/
An International Peer-Reviewed Journal

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14 Issue 12 , December - 2025

AutoML-Driven Flow-Level Classification of Encrypted Instant Messaging Applications

Ritwick Mondal
Department of Computer Science and Engineering
National Institute of Technology Durgapur
Durgapur, India

***Abstract*—The adoption of end-to-end encryption in Instant Messaging Applications (IMAs) has rendered payload-based traffic inspection ineffective, creating challenges for network monitoring, security enforcement, and quality-of-service management. Flow-level traffic classification has emerged as a practical alternative, but achieving high performance requires careful feature engineering and robust model optimization. This study presents an AutoML-driven framework for multiclass classification of encrypted IMA traffic using flow-level statistical features extracted with Tranalyzer2. A domain-driven feature filtering strategy removes protocol-specific and location-dependent attributes, yielding a refined set of 194 application-agnostic features, including the target class. We evaluate two state-of-the-art AutoML frameworks, AutoGluon and FLAML, on a dataset comprising six encrypted IMAs—Microsoft Teams, Discord, Facebook Messenger, Signal, Telegram, and WhatsApp—and four non-IMA traffic categories. AutoGluon constructs an ensemble model achieving 99.96% accuracy, whereas FLAML optimizes a LightGBM model within a limited time budget, achieving 99.94% accuracy with lower computational overhead. The comparative analysis highlights the trade-offs between ensemble-driven robustness and lightweight optimization efficiency. These results demonstrate that flow-level representations combined with modern AutoML frameworks enable highly accurate, scalable, and reproducible classification of encrypted IMA traffic while eliminating the need for manual model selection and hyperparameter tuning, making the approach suitable for deployment in real-world, resource-constrained network environments.**

***Index Terms*—Encrypted Traffic Classification; Instant Messaging Applications; AutoML; Flow-Level Features**

I. Introduction

The proliferation of end-to-end encryption in Instant Messaging Applications (IMAs) such as WhatsApp, Signal, Telegram, and Microsoft Teams has significantly enhanced user privacy but simultaneously challenged traditional network monitoring and traffic analysis methods. Payload-based inspection is rendered ineffective under encryption, limiting the ability of network administrators and security systems to accurately identify and manage application traffic. Consequently, flow-level traffic classification has emerged as a viable alternative, leveraging statistical and temporal characteristics of network flows rather than payload contents to distinguish between encrypted applications.

Accurate classification of encrypted IMA traffic remains a challenging task due to the high dimensionality and heterogeneity of flow-level features, as well as the presence of non-linear correlations between traffic characteristics and application behavior. Traditional machine learning approaches often require extensive manual feature engineering, model selection, and hyperparameter tuning, which can be labor-intensive, time-consuming, and prone to overfitting, particularly when datasets are large and complex. Moreover, certain features, such as protocol identifiers or five-tuple attributes, may trivially reveal application categories without capturing intrinsic traffic patterns, necessitating careful domain-driven preprocessing to ensure robust generalization.

Automated Machine Learning (AutoML) frameworks provide an effective solution to these challenges by streamlining model selection, feature preprocessing, and hyperparameter optimization in a reproducible and scalable manner. This study investigates the application of two state-of-the-art AutoML frameworks, AutoGluon [2] and FLAML [3], for multiclass classification of encrypted IMA traffic. Flow-level features are extracted using Tranalyzer2, a high-performance and extensible traffic analysis tool, and refined through domain-driven feature filtering to produce an application-agnostic feature set. AutoGluon leverages an ensemble-driven approach to achieve high robustness, while FLAML employs time-budgeted optimization for lightweight yet accurate model selection.

The contributions of this work are threefold: (i) the development of a robust AutoML-based framework for encrypted IMA traffic classification using flow-level features, (ii) a systematic evaluation of ensemble-driven versus lightweight AutoML strategies, and (iii) an empirical demonstration of near-perfect classification performance across multiple encrypted IMAs with minimal manual intervention, highlighting the suitability of the proposed approach for resource-constrained, real-world network environments.

II. Related Work

Machine learning has been widely adopted for network traffic analysis tasks such as application identification, encrypted traffic classification, malware detection, and intrusion detection. Early approaches relied heavily on manually engineered features and carefully tuned models, making them labor-intensive and difficult to scale. With the widespread adoption of encryption, recent research has shifted toward payload-agnostic representations and automated learning pipelines. Representative efforts include nPrintML [8], which integrates packet-level representations with AutoML to reduce manual feature engineering, and GGFAST [9], which constructs interpretable classifiers using packet size sequences and flow-level n-grams. While these methods demonstrate strong performance in payload-independent traffic analysis, they primarily operate on packet-level representations or task-specific classifiers and do not systematically evaluate general-purpose AutoML frameworks for application-level traffic classification.

More recent studies have explored AutoML and deep learning for encrypted traffic analysis. AutoML4ETC [10] applies neural architecture search to encrypted traffic classification using packet header information, while Isingizwe et al. [11] employ AutoML pipelines for encrypted malware traffic detection with automated model selection and ensemble learning. Although these works highlight the potential of AutoML in reducing tuning complexity and improving performance, their focus is largely limited to malware detection or packet-level classification, leaving encrypted Instant Messaging Application (IMA) traffic and flow-level statistical modeling relatively unexplored.

In contrast, this work addresses this gap by presenting a comprehensive AutoML-based framework for encrypted IMA traffic classification using flow-level statistical features extracted via Tranalyzer2 [1]. Unlike prior studies, we systematically evaluate both ensemble-driven and lightweight, time-budgeted AutoML frameworks—AutoGluon and FLAML—under identical experimental conditions, providing practical insights into performance–efficiency trade-offs. Section III details the proposed methodology and feature extraction process, followed by the rationale for selecting Tranalyzer2 in Section IV, AutoGluon-based experimental analysis in Section V, FLAML-based experimental analysis in Section VI, a comparative evaluation of AutoML frameworks in Section VII, and concluding insights in Section VIII.

III. Methodology

This section presents the complete experimental methodology adopted for encrypted Instant Messaging Application (IMA) traffic classification using AutoML frameworks. The proposed workflow is designed to ensure robustness, scalability, and reproducibility while avoiding information leakage and dataset bias. An overview of the end-to-end pipeline is illustrated in Figure 1.

The raw network traffic used in this study was obtained from the publicly available *Encrypted Mobile Instant Messaging Traffic Dataset* hosted on IEEE DataPort [4]. The dataset consists of ten packet capture (PCAP) files collected under controlled experimental conditions. Six PCAP files correspond to encrypted IMA traffic generated by Microsoft Teams, Discord, Facebook Messenger, Signal, Telegram, and WhatsApp, while the remaining four PCAP files represent non-IMA traffic, including web browsing, YouTube streaming, Gmail usage, and background system services. This composition enables the formulation of a realistic multi-class classification problem that reflects both application-specific encrypted communication and heterogeneous background traffic.

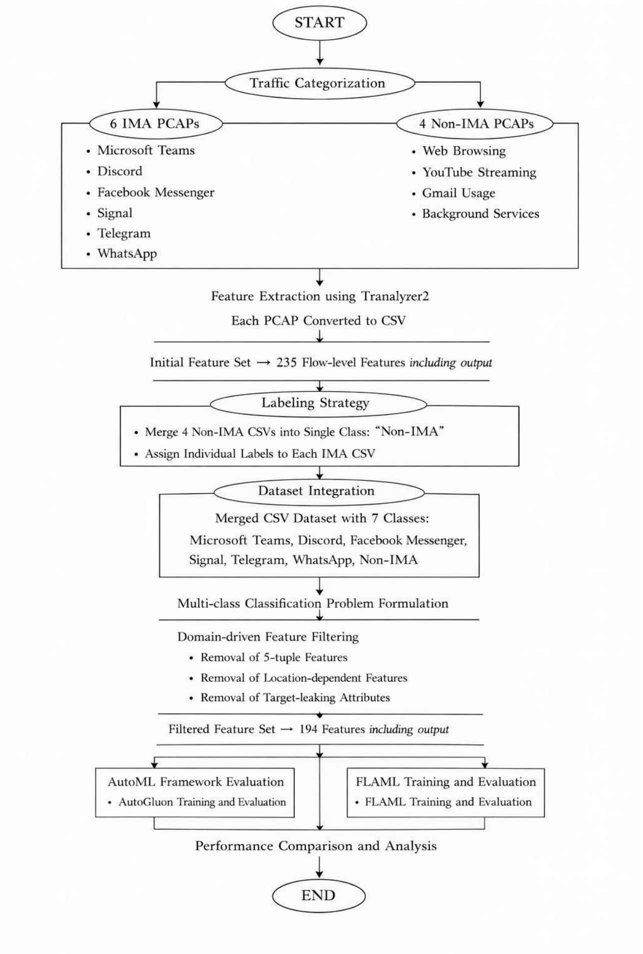
START
Traffic Categorization
6 IMA PCAPs
• Microsoft Teams
• Discord
• Facebook Messenger
• Signal
• Telegram
• WhatsApp
4 Non-IMA PCAPs
• Web Browsing
• YouTube Streaming
• Gmail Usage
• Background Services
Feature Extraction using Tranalyzer2
Each PCAP Converted to CSV
Initial Feature Set → 235 Flow-level Features including output
Labeling Strategy
• Merge 4 Non-IMA CSVs into Single Class: "Non-IMA"
• Assign Individual Labels to Each IMA CSV
Dataset Integration
Merged CSV Dataset with 7 Classes:
Microsoft Teams, Discord, Facebook Messenger,
Signal, Telegram, WhatsApp, Non-IMA
Multi-class Classification Problem Formulation
Domain-driven Feature Filtering
• Removal of 5-tuple Features
• Removal of Location-dependent Features
• Removal of Target-leaking Attributes
Filtered Feature Set → 194 Features including output
AutoML Framework Evaluation
• AutoGluon Training and Evaluation
FLAML Training and Evaluation
• FLAML Training and Evaluation
Performance Comparison and Analysis
END

Fig. 1: End-to-end workflow for encrypted IMA traffic classification, illustrating the processing pipeline from raw PCAP collection and flow-level feature extraction to AutoML-based model training and evaluation.

Each PCAP file was processed independently using Tranalyzer2, a high-performance flow-based traffic analysis framework. Tranalyzer2 aggregates packets into bidirectional flows and extracts a comprehensive set of statistical, temporal, transport-layer, entropy-based, and frequency-domain features via its modular plugin architecture. The output of each PCAP processing stage is a machine-learning-ready CSV file. In total, an initial feature set of 235 flow-level attributes, including the target label, was obtained. The extracted dataset comprises 106,284 flow records, providing sufficient scale and diversity for reliable supervised learning.

Following feature extraction, all CSV files were merged into a single integrated dataset. A domain-informed labeling strategy was applied to better reflect realistic deployment scenarios. Specifically, the four non-IMA traffic CSVs were consolidated into a single *Non-IMA* class, while each encrypted IMA application was assigned an individual class label. This resulted in a seven-class classification problem consisting of six encrypted IMA classes and one aggregated non-IMA class.

To enhance generalization performance and prevent information leakage, a domain-driven feature filtering process was conducted prior to model training. Features that could trivially reveal traffic identity without capturing intrinsic behavioral characteristics were removed. These include protocol identifiers, five-tuple attributes (source IP address, destination IP address, source port, destination port, and transport-layer protocol), as well as location-dependent features. Such attributes may artificially inflate classification performance while reducing robustness in real-world scenarios. After filtering, a refined feature set of 194 flow-level attributes, including the target label, was retained. These features collectively provide an application-agnostic representation of encrypted traffic suitable for automated learning.

The refined dataset was randomly split into training and testing subsets using an 80:20 ratio, resulting in 85,027 training instances and 21,257 testing instances. The classification task was formulated as a multi-class supervised learning problem. Two state-of-the-art AutoML frameworks, AutoGluon and FLAML, were employed to evaluate automated model selection, feature handling, and hyperparameter optimization under identical experimental conditions.

AutoGluon-Tabular was applied directly to the filtered dataset without manual preprocessing. The framework automatically inferred feature types, handled missing values, encoded categorical attributes, and removed non-informative features. During training, AutoGluon evaluated a diverse pool of learners, including gradient boosting models, ensemble tree methods, and neural networks, and combined them using a weighted ensemble strategy. The default medium preset was used to balance computational efficiency and predictive performance, leading to the automatic selection of the `WeightedEnsemble_L2` model as the final predictor.

To further assess the effectiveness of lightweight AutoML under strict time constraints, FLAML was evaluated on the same dataset and train–test split. FLAML was configured for multiclass classification with a fixed time budget of 120 seconds and optimized the log-loss objective. During optimization, FLAML dynamically allocated computational resources across candidate learners, including LightGBM, Random Forest, Extra Trees, XGBoost, and linear models, ultimately converging to an optimized LightGBM classifier.

Finally, the performance of AutoGluon and FLAML was systematically compared using standard evaluation metrics, including accuracy, balanced accuracy, precision, recall, and F1-score. This comparative analysis enables a rigorous assessment of the trade-offs between predictive performance, automation capability, and computational efficiency in the context of encrypted IMA traffic classification.

IV. Rationale for Selecting Tranalyzer2

Based on a systematic comparison of widely used open-source network traffic analysis frameworks, Tranalyzer2 was selected as the flow-level feature extraction tool for this study due to its strong alignment with the requirements of encrypted Instant Messaging Application (IMA) traffic classification. Unlike packet-centric analysis tools such as Scapy [6], or flow generators with comparatively limited statistical expressiveness such as CICFlowMeter [5], Tranalyzer2 natively produces machine-learning-ready CSV outputs enriched with a comprehensive and extensible set of flow-level attributes through its modular plugin architecture.

In contrast to Zeek [7], which primarily focuses on semantic and event-driven logging and often requires substantial post-processing to adapt outputs for machine learning workflows, Tranalyzer2 directly exposes a wide spectrum of discriminative features encompassing statistical, temporal, transport-layer, and TLS-related characteristics. This enables fine-grained behavioral modeling of encrypted traffic without reliance on payload inspection. Its support for multiple protocols, including TCP, UDP, SSL/TLS, and application-aware deep packet inspection (DPI) plugins, is particularly well suited for modern IMA traffic, where encryption renders content-based analysis infeasible.

From a scalability perspective, Tranalyzer2 is optimized for high-throughput PCAP processing and demonstrates strong performance when handling large-scale traffic traces, a critical requirement for flow-based learning tasks. Furthermore, its lightweight and modular design allows controlled extensibility via plugins without introducing scripting overhead or complex configurations, in contrast to frameworks that depend heavily on custom scripts. Collectively, these characteristics motivate the choice of Tranalyzer2 for flow-level feature extraction, as it provides a robust, scalable, and encryption-resilient representation of network traffic that integrates seamlessly with the automated machine learning pipelines employed in this work.

V. AutoGluon-Based Experimental Analysis

This section presents a detailed experimental analysis of the proposed encrypted Instant Messaging Application (IMA) traffic classification framework using AutoGluon-Tabular. Following an 80:20 train–test split, the resulting training and testing datasets comprised 85,027 and 21,257 instances, respectively, each represented by 194 flow-level features including target class. AutoGluon was employed for multiclass classification with accuracy as the primary optimization metric, automatically formulating a seven-class prediction problem corresponding to six encrypted IMA applications and a consolidated non-IMA class.

During preprocessing, AutoGluon's automated feature engineering pipeline performed data type inference, missing value handling, categorical encoding, and feature transformation, while identifying and excluding non-informative attributes. Specifically, 16 features were removed due to constant values, and an additional 25 features were ignored owing to limited informational contribution. As a result, 152 original features were retained and transformed into 161 processed features, reducing memory usage while preserving discriminative capability. A summary of the feature composition before and after preprocessing is provided in Table II.

TABLE I: Model-wise Performance Metrics (Rounded to 3 Decimal Places)

Model	Acc.	Bal. Acc.	Prec. M	Prec. μ	Prec. W	Rec. M	Rec. μ	Rec. W	F1 M	F1 μ	F1 W
XGBoost	0.999	0.999	0.999	0.999	0.999	0.999	0.999	0.999	0.999	0.999	0.999
RandomForest (Entropy)	0.999	0.999	0.999	0.999	0.999	0.999	0.999	0.999	0.999	0.999	0.999
RandomForest (Gini)	0.999	0.999	0.999	0.999	0.999	0.999	0.999	0.999	0.999	0.999	0.999
ExtraTrees (Gini)	0.999	0.999	0.999	0.999	0.999	0.999	0.999	0.999	0.999	0.999	0.999
ExtraTrees (Entropy)	0.999	0.999	0.999	0.999	0.999	0.999	0.999	0.999	0.999	0.999	0.999
NeuralNetFastAI	0.999	0.998	0.998	0.999	0.999	0.998	0.999	0.999	0.998	0.999	0.999
WeightedEnsemble L2	0.999	0.998	0.998	0.999	0.999	0.998	0.999	0.999	0.998	0.999	0.999
LightGBM-XT	0.999	0.998	0.998	0.999	0.999	0.998	0.999	0.999	0.998	0.999	0.999
NeuralNetTorch	0.998	0.998	0.998	0.998	0.998	0.998	0.998	0.998	0.998	0.998	0.998
LightGBM	0.998	0.998	0.998	0.998	0.998	0.998	0.998	0.998	0.998	0.998	0.998
LightGBM-Large	0.998	0.997	0.999	0.998	0.998	0.997	0.998	0.998	0.998	0.998	0.998
CatBoost	0.996	0.995	0.994	0.996	0.996	0.995	0.996	0.996	0.995	0.996	0.996

TABLE II: Summary of Feature Types Used by AutoGluon

Feature Category	Count
Total Original Features	193
Useless Features (Constant)	16
Unused Features (Ignored)	25
Original Features Considered	152
Processed Features Generated	161
Integer Features	62
Boolean Features	7
Float Features	53
Categorical Features	27
Datetime-derived Features	12

During the AutoML training phase, a diverse pool of base learners was evaluated, including gradient boosting models (LightGBM, LightGBMXT, LightGBMLarge, and XGBoost), ensemble tree-based methods (Random Forest and Extra Trees with both Gini and Entropy criteria), neural network architectures (NeuralNetFastAI and NeuralNetTorch), and CatBoost. Several models achieved validation accuracies exceeding 99.8%. Based on validation performance, the `WeightedEnsemble_L2` model was automatically selected as the final predictor, with the NeuralNetFastAI model contributing most significantly to the ensemble. As no explicit preset was specified, AutoGluon defaulted to the *medium* preset, offering a balanced trade-off between computational efficiency and predictive performance.

A detailed comparison of model-wise performance metrics on the held-out test set is reported in Table I. The consistently high accuracy, balanced accuracy, precision, recall, and F1-scores across all evaluated models indicate strong separability of encrypted IMA traffic when represented using flow-level statistical features.

To enhance interpretability, feature importance scores were extracted from the trained AutoGluon ensemble. Figure 2 highlights the most influential flow-level attributes contributing to encrypted IMA traffic discrimination, demonstrating that the classifier relies primarily on intrinsic traffic dynamics rather than protocol identifiers or payload-dependent characteristics. The feature importance analysis indicates that AutoGluon primarily leverages temporal and flow-structural characteristics for encrypted IMA traffic classification. Highly ranked features such as `nDPIclass`, `timeFirst`, and `timeLast` highlight the significance of coarse protocol cues and flow timing information. Connection-level attributes including `connSip`, `connDip`, and `connSipDptr` further emphasize the role of source–destination interaction patterns. Overall, the dominance of payload-independent statistical features confirms that robust encrypted traffic discrimination can be achieved using intrinsic traffic dynamics without relying on payload inspection.

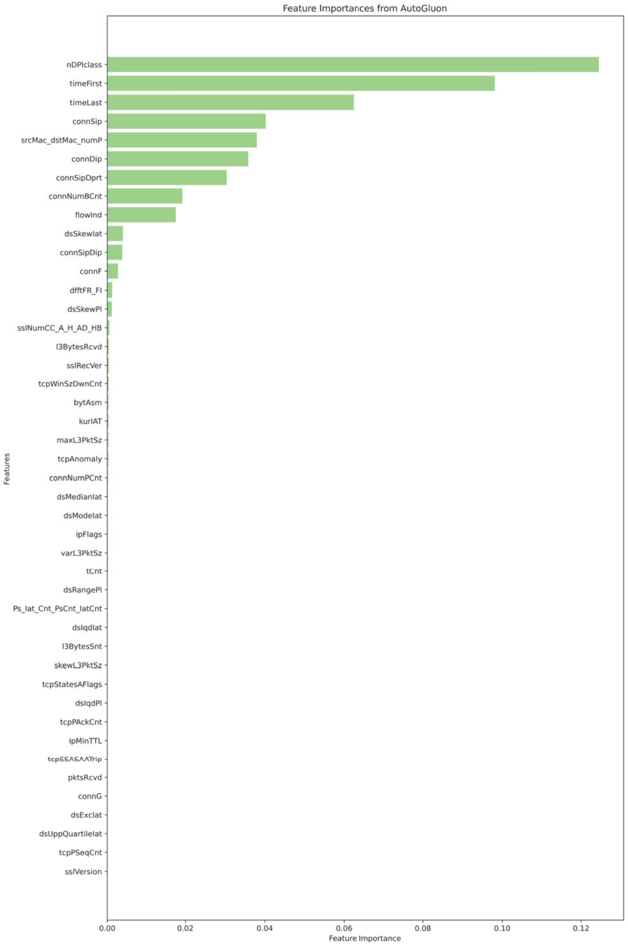

Fig. 2: Top-ranked flow-level feature importances obtained from the AutoGluon ensemble model.

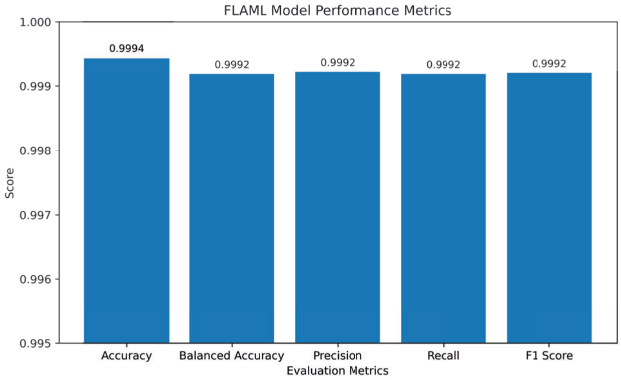

Fig. 3: Performance metrics of the optimized FLAML model on the encrypted IMA traffic classification task.

VI. FLAML-Based Experimental Analysis

To further investigate the effectiveness of lightweight AutoML frameworks for encrypted Instant Messaging Application (IMA) traffic classification, additional experiments were conducted using FLAML (Fast Lightweight AutoML). The experiments were performed on the same dataset and refined feature space used in the AutoGluon analysis to ensure a fair and controlled comparison. Specifically, the complete set of 194 flow-level features including target class, obtained after domain-driven filtering, was employed along with an identical 80:20 train–test split.

FLAML was configured for multiclass classification with a fixed time budget of 120 seconds and log-loss as the optimization objective. Under this setting, FLAML dynamically allocated computational resources across a diverse pool of candidate learners, including LightGBM, Random Forest, Extra Trees, XGBoost, depth-limited XGBoost, stochastic gradient descent (SGD), and L1-regularized logistic regression. The optimization process followed a holdout validation strategy, progressively refining hyperparameters while accounting for heterogeneous evaluation costs across models.

Throughout the optimization process, LightGBM consistently emerged as the most competitive learner. As the available time budget increased, FLAML explored more expressive configurations of LightGBM, ultimately converging to an optimized classifier that achieved the lowest validation error among all evaluated models. The final retrained LightGBM model obtained under the 120-second budget delivered near-perfect classification performance on the held-out test set.

The quantitative performance of the optimized FLAML model is illustrated in Figure 3. The model achieved an accuracy of 99.94%, balanced accuracy of 99.92%, precision of 99.92%, recall of 99.92%, and an F1-score of 99.92%. These results demonstrate that FLAML is able to extract strong predictive performance from flow-level encrypted traffic features while maintaining significantly lower computational overhead compared to ensemble-heavy AutoML frameworks.

To enhance model interpretability, feature importance scores were extracted from the optimized LightGBM classifier produced by FLAML. Figure 4 presents the top-ranked flow-level attributes contributing to the classification decision. The results indicate that connection-level, temporal, and statistical flow descriptors dominate the decision-making process, confirming that discriminative information is primarily derived from intrinsic traffic dynamics rather than payload content or explicit protocol identifiers.

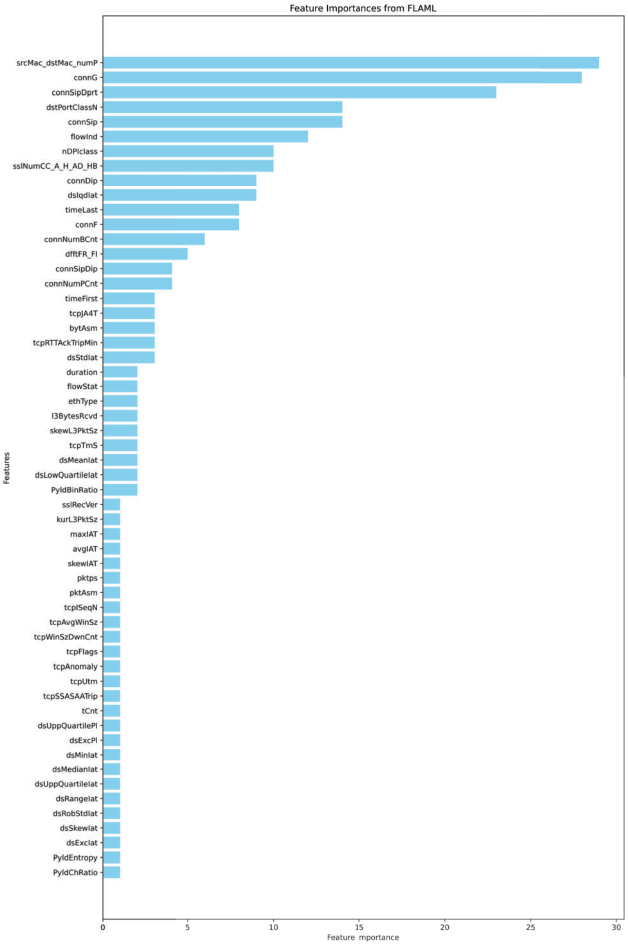

Fig. 4: Top-ranked flow-level feature importances obtained from the FLAML-optimized LightGBM model.

Features such as `srcMac_dstMac_numP`, `connG`, and `connSipDprt` emerge as the top contributors, highlighting the importance of communication patterns, connection grouping behavior, and destination port dynamics in distinguishing encrypted IMA traffic. These features capture how endpoints interact over time and how frequently specific communication pairs exchange packets, which are strong indicators of application-specific behavior even under encryption. Similarly, the prominence of `dstPortClassN`, `connSip`, and `flowInd` suggests that port class abstraction and flow indexing information play a crucial role in characterizing traffic without exposing sensitive payload information.

Temporal attributes such as timeLast, timeFirst, and inter-arrival-time-related metrics (e.g., avgIAT, maxIAT) further reinforce the observation that timing behavior is a key discriminative factor. These features capture session duration, burstiness, and packet spacing patterns, which tend to differ significantly across instant messaging applications due to their underlying communication protocols and user interaction models. In addition, statistical flow descriptors, including byte and packet asymmetry metrics (e.g., bytAsm, l3BytesRcvd), provide insight into data exchange balance between sender and receiver, further strengthening class separability.

Lower-ranked features, such as TCP window statistics and payload entropy-related attributes, contribute marginally to the final decision. This gradual decline in importance demonstrates that while these features add contextual information, the core predictive power is concentrated in a relatively compact subset of flow-level characteristics. Importantly, the absence of heavy reliance on payload entropy or protocol identifiers confirms the robustness of the approach in encrypted settings, where deep packet inspection is infeasible.

Overall, this detailed feature-importance analysis substantiates that FLAML not only achieves near-perfect classification performance under strict time constraints but also yields interpretable and behavior-driven models. The dominance of connection-level, temporal, and statistical features validates the effectiveness of flow-based analysis for encrypted IMA traffic classification. Consequently, FLAML proves to be an efficient, scalable, and practical AutoML framework for real-time and resource-constrained network monitoring environments, where rapid optimization, high accuracy, and transparent decision-making are essential.

VII. Comparative Analysis of AutoGluon and FLAML AutoML Frameworks

While both AutoGluon and FLAML are designed to automate model selection and hyperparameter optimization, they differ substantially in design philosophy, computational strategy, and practical deployment objectives. To provide a clearer methodological perspective, Table III presents a structured comparison between the two frameworks as used in this study.

TABLE III: Comparison between FLAML and AutoGluon

Feature / Aspect	**FLAML**	**AutoGluon**
Purpose	Lightweight AutoML for fast hyperparameter optimization	Comprehensive AutoML framework for high-performance modeling
Primary Focus	Time- and resource-efficient model selection	Maximizing predictive performance via ensembles
Supported Tasks	Classification, regression, custom function tuning	Classification, regression, and multimodal learning
Base Learners	LightGBM, XGBoost, Random Forest, Extra Trees, linear models	LightGBM, CatBoost, XGBoost, neural networks
Hyperparameter Tuning	Budget-aware, cost-frugal optimization with early stopping	Bayesian optimization, grid search, and meta-learning
Ensembling Strategy	Limited; focuses on selecting the best model	Extensive stacking and bagging
Ease of Use	Simple, scikit-learn–style interface	Feature-rich with moderate complexity
Speed / Resource Usage	Very fast with low memory footprint	Slower with higher memory and compute requirements
Zero-Shot AutoML	Supported	Not supported
Best Use Case	Rapid experimentation and resource-limited deployment	Achieving near state-of-the-art accuracy

FLAML is explicitly optimized for efficiency under strict time and resource constraints. Its budget-aware optimization strategy dynamically allocates computational effort toward promising configurations, enabling rapid convergence to high-performing models with minimal overhead. This makes FLAML particularly suitable for time-sensitive and resource-constrained environments, such as real-time encrypted network traffic analysis. Moreover, its lightweight design ensures that it can be easily integrated into automated pipelines without requiring specialized hardware, allowing for scalable deployment across multiple devices or network nodes. Its ability to quickly adapt to changing data patterns further reinforces its utility in dynamic traffic scenarios where rapid decision-making is essential.

In contrast, AutoGluon prioritizes predictive performance through extensive ensembling, stacking, and bagging strategies. By leveraging a diverse set of base learners and multi-level ensemble construction, AutoGluon is able to achieve strong generalization performance, albeit at the cost of increased training time and memory usage. This design makes AutoGluon well suited for scenarios where computational resources are less constrained and maximizing classification accuracy is the primary objective. Additionally, its built-in support for multi-modal data and automated feature engineering further enhances its versatility in complex traffic classification tasks. Its robustness to noisy or incomplete datasets also ensures consistent performance across diverse network environments.

Overall, the comparison highlights a fundamental trade-off between computational efficiency and ensemble-driven performance. While both frameworks demonstrate strong suitability for encrypted IMA traffic classification, FLAML offers faster optimization and lower resource consumption, whereas AutoGluon provides a more comprehensive and performance-oriented AutoML pipeline. The complementary strengths of these frameworks justify their joint evaluation in this work. Understanding these differences also provides valuable guidance for practitioners when selecting an AutoML tool based on specific project constraints, operational environments, and desired outcomes. By carefully considering the trade-offs between speed, resource usage, and predictive accuracy, researchers and engineers can make informed decisions that align with both experimental and real-world deployment goals.

VIII. Conclusion

This study demonstrates that encrypted Instant Messaging Application (IMA) traffic can be accurately classified using flow-level statistical features without reliance on payload inspection or protocol-specific identifiers. By employing Tranalyzer2 for feature extraction, the proposed framework effectively captures intrinsic traffic dynamics that remain robust under encryption.

A systematic evaluation of two AutoML frameworks highlights their complementary strengths. AutoGluon achieves near-optimal classification performance through ensemble-driven optimization, making it well suited for scenarios where maximum predictive performance and generalization are required. In contrast, FLAML delivers comparable performance with significantly lower computational cost, emphasizing its suitability for time-critical and resource-constrained environments.

Overall, the results confirm that AutoML provides a scalable and reproducible solution for encrypted traffic classification, enabling informed trade-offs between accuracy and efficiency. These findings offer practical guidance for deploying automated learning systems in real-world, privacy-preserving network monitoring applications.

References

[1] Tranalyzer2 Documentation. Available at: https://tranalyzer.com/documentation

[2] AutoGluon: AutoML for Tabular, Text, and Image Data. Available at: https://auto.gluon.ai/stable/index.html

[3] Microsoft FLAML: Fast Lightweight AutoML. Available at: https://microsoft.github.io/FLAML/

[4] Encrypted Mobile Instant Messaging Traffic Dataset. IEEE DataPort. Available at: https://ieee-dataport.org/documents/encrypted-mobile-instant-messaging-traffic-dataset

[5] Ahlashkari, A., et al.: CICFlowMeter: A Network Traffic Flow Generator. Available at: https://github.com/ahlashkari/CICFlowMeter

[6] Scapy: Packet Manipulation Program. Available at: https://scapy.net/

[7] Zeek Network Security Monitor. Available at: https://zeek.org/

[8] Holland, J., Schmitt, P., Feamster, N., Mittal, P.: New Directions in Automated Traffic Analysis. In: *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, pp. 3366–3383. ACM (2021). DOI: https://doi.org/10.1145/3460120.3484758

[9] Piet, J., Nwoji, D., Paxson, V.: GGFAST: Automating Generation of Flexible Network Traffic Classifiers. In: *Proceedings of the ACM SIGCOMM 2023 Conference*, pp. 850–866 (2023). DOI: https://doi.org/10.1145/3603269.3604840

[10] Malekghaini, N., et al.: AutoML4ETC: Automated Neural Architecture Search for Real-World Encrypted Traffic Classification. *IEEE Transactions on Network and Service Management*, 21(3), 2715–2730 (2024). DOI: https://doi.org/10.1109/TNSM.2023.3324936

[11] Isingizwe, D.F., Wang, M., Liu, W., Wang, D., Wu, T., Li, J.: Analyzing Learning-based Encrypted Malware Traffic Classification with AutoML. In: *2021 IEEE 21st International Conference on Communication Technology (ICCT)*, pp. 313–322. IEEE (2021). DOI: https://doi.org/10.1109/ICCT52962.2021.9658106