**PREDICTING TOR TRAFFIC CLASSIFICATION FROM SECURED PAYLOAD USING MACHINE LEARNING**

1K.LAVANYA, 2K.RICHITHA, 3K.HARSHITHA, 4G.SENTHILVELAN, 5S.RAMANUJAM, 6V.B. GANAPATHY

1,2,3Student, 4Assistant Professor, 5,6Professor

Department of Computer Science and Engineering, Dr. MGR Educational And Research Institute

Maduravoyal, Chennai, Tamil Nadu

E-mail:1 lavanyareddy2106@gmail.com, 2 richitha14@gmail.com, 3 harshitha.kurra2003@gmail.com

4senthilvelan.cse@drmgrdu.ac.in,5ramanujam.cse@drmgrdu.ac.in,6ganapathy.cse@drmgrdu.ac.in

*Abstract*— The rapid evolution of internet technologies has necessitated advanced methodologies for monitoring and classifying encrypted network traffic. This study introduces a robust framework utilizing Machine Learning (ML) to classify Tor traffic encrypted payloads, an essential step for enhancing cybersecurity measures. Utilizing a dataset featuring columns such as Source Port, Destination Port, Protocol, Flow Duration, various Inter-Arrival Times (IAT), and others, we apply three distinct ML models: Decision Tree, Logistic Regression, and XG tor boost. Our objective is to accurately predict the nature of traffic ('label' as the target column), thereby distinguishing between benign and potentially malicious activities. The effectiveness of each model is evaluated based on their predictive accuracy and computational efficiency, offering insights into the optimal approaches for real-time encrypted traffic analysis. This research contributes to the development of more secure network environments. Our research makes a valuable contribution to network security by proposing an innovative and efficient approach for identifying Tor traffic. This study seeks to bridge the existing gap by introducing a new method that effectively addresses these limitations..

Keywords—Decision Tree, Logistic Regression,XG tor boost, Random Forest, Tor Traffic Classification.

# **Introduction**

Tor is an anonymous network designed to enhance user privacy by masking their identities. It achieves this by directing traffic through multiple relays distributed across the Tor network, which is operated by volunteers worldwide. This multi-layered routing mechanism effectively obscures the origin of Tor traffic, making it challenging to trace users' actual IP addresses. Such anonymity benefits a diverse group of users, including individuals seeking to avoid ISP surveillance, as well as journalists and activists who require secure and private communication channels.

Our research makes a valuable contribution to network security by introducing an innovative and efficient approach to detecting Tor traffic. This study aims to bridge existing gaps by proposing a novel method that addresses current limitations in Tor traffic detection.

The distinctive feature of Tor lies in its multilayered encryption, which enhances privacy significantly compared to the single-layer encryption used in non-Tor networks., has captured our attention. The application offers a user-friendly interface where patients can register, authenticate, and interact with medical professionals. Key functionalities include selecting suitable healthcare providers, viewing detailed doctor profiles, and scheduling appointments—all through an intuitive design that enhances accessibility and usability.

The rise of encrypted network traffic has brought both significant privacy benefits and new challenges in network security. One of the most widely used privacy-focused networks is the Onion Router (Tor), a platform that anonymizes user activities and prevents traffic analysis by routing connections through a series of distributed nodes. Tor encrypts data multiple times, ensuring user anonymity and data confidentiality, but this also complicates the work of cybersecurity professionals tasked with identifying and mitigating malicious activities that may exploit Tor’s privacy features. This approach leverages patterns and statistical features, rather than payload content, allowing for non-invasive classification methods that respect user privacy..

This study aims to explore machine learning algorithms to classify Tor traffic based on network metadata, such as packet size, timing, direction, and flow. By applying various models and evaluating their accuracy, this research will contribute to understanding the effectiveness of machine learning in detecting Tor traffic while preserving user anonymity and privacy.

# **Related Works**

Several studies have explored machine learning approaches for classifying encrypted Tor traffic, providing valuable insights into the challenges and advancements in this domain. Johnson et al. (2021) investigated deep learning methods for characterizing Tor traffic using time-based features and achieved over 99% accuracy in detecting Tor traffic. Similarly, Salman et al. (2020) reviewed various machine learning-based Internet traffic classification techniques, emphasizing their advantages over traditional port-based and deep packet inspection methods. Rezaei and Liu (2019) provided an overview of deep learning for encrypted traffic classification, discussing the evolution of classification techniques and highlighting the effectiveness of neural networks in handling encrypted data.

Choorod and Weir (2021) introduced a novel approach that utilized deep packet inspection and machine learning to classify Tor traffic based on encrypted payload characteristics, achieving a classification accuracy of 95.65%. Sarkar et al. (2020) proposed a deep neural network-based system for detecting encrypted Tor traffic, reporting an impressive accuracy of 99.89%. Additionally, Lashkari et al. (2017) focused on the characterization of Tor traffic using time-based features, reinforcing the importance of statistical traffic analysis in identifying patterns within encrypted communications. Cuzzocrea et al. (2017) explored machine learning-based Tor traffic detection and emphasized the role of feature engineering in improving classification performance. These studies collectively provide a strong foundation for the current research, demonstrating the potential of machine learning in enhancing network security and encrypted traffic analysis. The insights from previous works guide the development of improved classification models, ensuring real-time detection and monitoring of encrypted network traffic while balancing privacy concerns.

# **Methodology**

The proposed system employs machine learning techniques to classify encrypted Tor traffic payloads, enhancing cybersecurity by distinguishing between benign and malicious activities. A dataset containing features such as Source Port, Destination Port, Protocol, Flow Duration, and Inter-Arrival Times (IAT) is used for analysis. The methodology involves data preprocessing, model selection, training, and evaluation.Several machine learning models are implemented for classification, including Decision Tree, Logistic Regression, XG tor boost, Random Forest, and Convolutional Neural Networks (CNNs). Decision Tree and Random Forest provide interpretability and feature importance analysis, while Logistic Regression is used for binary classification. XG tor boost, an optimized gradient boosting algorithm, enhances predictive performance and handles high-dimensional data efficiently. CNNs are explored for their hierarchical feature extraction capability, enabling them to analyze complex data structures within encrypted traffic.

***Decision Tree:***

Decision Tree models are pivotal In utilizing machine learning for the classification of encrypted Tor traffic payloads. They excel in analyzing complex, encrypted data patterns by recursively partitioning the data based on features such as payload size, timing characteristics, and metadata. This approach allows Decision Trees to effectively distinguish between different types of Tor traffic, including normal user activities and potentially malicious communications. By constructing a tree-like structure of decision rules, these models enable interpretable insights into the underlying patterns of encrypted payloads, aiding in the identification of anomalous or suspicious traffic. Leveraging machine learning techniques, Decision Trees contribute significantly to enhancing cybersecurity measures by providing automated and accurate classification of encrypted Tor network traffic, thereby bolstering network security and ensuring robust protection against various cyber threats**.**The process of splitting the data continues until certain stopping criteria are met. These criteria can include factors like a maximum depth of the tree, A minimum threshold of samples per leaf node is required, or further splits are halted when they no longer enhance the model’s predictive performance.The result is a tree structure that classifies or predicts outcomes based on the features used in the dataset.

However, decision trees have some drawbacks. They are prone to overfitting, especially when the tree becomes very deep and complex. In such cases, the tree may start to memorize the data, capturing noise and irrelevant patterns rather than general trends, which can reduce the model’s ability to generalize to new data. Overfitting occurs when the model becomes too complex, fitting every detail of the training data, including anomalies or noise.To combat overfitting, several techniques are commonly used. One method is pruning, which involves cutting back the tree after it has been fully grown to remove branches that do not contribute significantly to the model’s accuracy. Another approach is to limit the maximum depth of the tree or set a minimum number of samples per leaf, ensuring the tree does not become too deep and complex. Additionally, ensemble methods, such as Random Forest and Gradient Boosting, are often employed to improve the performance of decision trees. These techniques combine multiple decision trees to create a more robust and generalized model that mitigates the risk of overfitting.

***Logistic Regression:***

Logistic Regression is a fundamental machine learning algorithm suitable for classifying Tor traffic encrypted payloads. It operates by modeling the probability of categorical outcomes based on predictor variables, making it adept for binary classification tasks like distinguishing encrypted Tor traffic from non-Tor traffic. In this context, Logistic Regression analyzes features extracted from encrypted payloads, such as packet size distributions, traffic patterns, and encryption metadata. By learning a decision boundary between Tor and non-Tor traffic, it effectively identifies encrypted Tor packets with high accuracy. Leveraging its simplicity, interpretability, and ability to handle large datasets efficiently, Logistic Regression serves as a robust tool in the realm of network traffic analysis and cybersecurity,contributing to the detection and classification of encrypted Tor traffic with precision and reliability.The model is trained by maximizing the likelihood of observing the given data, which is done using optimization techniques like gradient descent. The coefficients β are adjusted to minimize the error between the predicted probabilities and the actual outcomes in the training data, typically using a cost function like cross-entropy loss. One of the main advantages of logistic regression is its simplicity and interpretability.

***XG tor boost Classifier:***

For classifying Tor traffic encrypted payloads using the XG tor boost classifier, the approach leverages its ability to handle complex, nonlinear relationships within data. XG tor boost, an optimized gradient boosting algorithm, excels in feature importance estimation and model interpretability, crucial for understanding encrypted payload patterns. By preprocessing data to extract relevant features like packet size, frequency, and timing characteristics, XG tor boost can effectively learn and classify Tor traffic. Its ensemble-based nature integrates decision trees sequentially, minimizing bias and variance while enhancing predictive performance. This model's robustness in handling high-dimensional, encrypted data makes it suitable for identifying anomalous or suspicious activities within Tor network traffic, aiding in cybersecurity efforts and network monitoring.Its ability to produce accurate models quickly, especially on structured data, has made it the go-to algorithm for data scientists and machine learning practitioners. In summary, XG tor boost is an extremely efficient and scalable machine learning algorithm that combines the power of gradient boosting with regularization, parallelization, and sophisticated handling of missing data. Its strong predictive performance, adaptability through custom objective functions, and ability to handle complex and large datasets make it an excellent choice for a wide range of tasks. Although it requires careful tuning and can be resource-intensive, its accuracy and robustness have made it a dominant algorithm in fields such as finance, e-commerce, healthcare, and marketing, where high accuracy and predictive power are critical for decision-making.

***Random Forest:*** Random Forest is a robust machine learning algorithm suitable for classifying encrypted payloads in Tor traffic. It excels in this context by leveraging an ensemble of decision trees, each trained on different subsets of features and data samples. For the classification of encrypted Tor traffic payloads, Random Forest offers several advantages: it handles high-dimensional data well, mitigates overfitting, and provides insight into feature importance, crucial for understanding which aspects of the payload contribute most to classification accuracy. By aggregating predictions from multiple decision trees, Random Forest improves robustness against noise and variability in encrypted data, making it an effective choice for identifying patterns and anomalies within Tor network traffic. Its ability to handle complex, encrypted data structures while maintaining interpretability makes Random Forest a valuable tool in the realm of cybersecurity and network traffic analysis.

***Convolutional Neural Networks (CNNs):***

Convolutional Neural Networks (CNNs) are pivotal in the classification of Tor traffic encrypted payloads using machine learning. By leveraging their hierarchical feature extraction capabilities, CNNs excel in analyzing complex data structures like encrypted traffic payloads. They operate by applying convolutional filters across input data to automatically learn spatial hierarchies of features, which is crucial for discerning patterns within encrypted packets. This approach enables CNNs to effectively differentiate between various types of Tor traffic based on subtle differences in payload structures and content. Furthermore, CNNs can adapt to diverse and evolving encryption schemes, making them versatile for real-time classification tasks in cybersecurity. Integrating CNNs into Tor traffic analysis enhances detection accuracy and efficiency, offering robust capabilities for identifying and mitigating potential threats within encrypted communication channels.

# 

*Fig1: System architecture for classifying tor traffic*

System architecture is the foundational structure of a software or hardware system, defining its components, interactions, and design principles to meet both functional and non-functional requirements. It typically includes multiple layers: the presentation layer (UI/UX) for user interactions, the business logic layer for processing and enforcing rules, the data access layer for database interactions, and the data storage layer where data is persistently held.

System architecture is the foundational structure of a software or hardware system, defining its components, interactions, and design principles to meet both functional and non-functional requirements. It typically includes multiple layers: the presentation layer (UI/UX) for user interactions, the business logic layer for processing and enforcing rules, the data access layer for database interactions, and the data storage layer where data is persistently held. Key architectural styles vary, such as monolithic architectures, where all components are tightly integrated, microservices architectures, which break the system into independent, loosely coupled services, and event-driven architectures, where components communicate asynchronously via events. Modern architectures often rely on APIs and web services for communication, load balancers for efficient request distribution, and containers for consistent deployment across environments.

Effective data management requires thoughtful database architecture and ETL pipelines to manage data flow, along with robust security layers for authentication, encryption, and compliance. Scalability and fault tolerance are essential, achieved through horizontal and vertical scaling, redundancy, and failover mechanisms. Systems also require monitoring and logging for real-time performance insights and error tracking, using tools like Prometheus or ELK. Additionally, CI/CD pipelines automate testing and deployment, supporting faster, reliable updates.

**IV.EXPERIMENTAL ANALYSIS**

**AND RESULT**

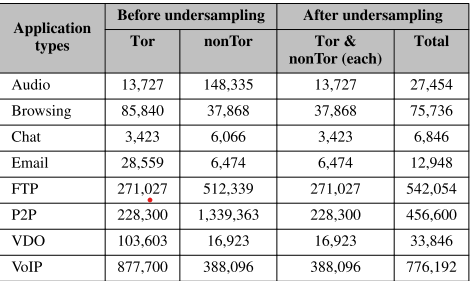
1) **DATA CLEANSING:** In today's digital environment, numerous applications employ dynamic ports as a method to evade network restrictions or surveillance.However, even commonly associated ports may not accurately identify a specific application. For instance, Tor traffic can operate on port 443, a port typically linked to HTTPS. To ensure precise data extraction, we closely follow the dataset descriptions specified in this study.

Data Cleansing Algorithm:

**Input:** F - Network flow in PCAP format  
**Output:** E - Encrypted payload

1. Initialize i = 0 (starting with the first packet in the PCAP file).
2. While i < F, perform the following steps:
   * Read packet i.
   * If the topmost layer protocol is "TLS":
     + Extract payload i.
     + If payload i is non-empty, append tls.app\_data to E.
   * Else if the protocol is "SSH":
     + Extract payload i.
     + If payload i is non-empty, append ssh.encrypted\_packet to E.
   * Else if the protocol is "TCP" or a proprietary protocol:
     + Extract payload i.
     + If payload i is non-empty, append tcp.data to E.
   * Increment i by 1.
3. Return E.

2) FEATURE EXTRACTION:The extracted features were categorized into two statistical sets: frequency and ratio calculations, obtained using the mathematical formulas defined below.



***Table 1*** *presents the count of Tor and non-Tor encrypted payloads across all application types, both before and after applying data balancing techniques.*

Feature Extraction Algorithm

**Input:** P-Extracted encrypted payloads  
**Output:** F - Frequency count of individual hex characters

R - Ratio of hex character frequency in payloads

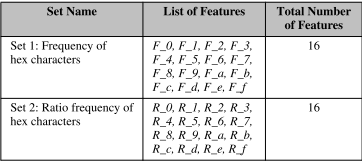
1: For each packet in P:

2: If the packet contains ",", split it into new rows.

3: Compute the frequency count (F).

4: Compute the ratio of frequency values (R).

5: return F, R

Set 2 comprises the 16 hexadecimal character frequency ratio features (R\_0 - R\_f), while (F\_0 - F\_f) represents the initial feature set

*TABLE 2. List of features used in the analyses.*

**A. STATISTICAL ANALYSIS**

Inferential statistics help determine patterns in data samples to infer conclusions about larger populations. This study employs the Mann-Whitney U test, a nonparametric method, to assess whether Tor and non-Tor encrypted payloads exhibit significant differences, particularly in cases where data distribution is not normal.The calculation of the Mann-Whitney Utest statistic as follows:



where:

* U represents the test statistic.
* U1 and U2 are the rank sums for Tor and non-Tor payloads.
* n1 and n2 are the sample sizes for Tor and non-Tor payloads.
* Ri and Rj represent ranked observations across both samples.

.

A p-value lower than 0.05 suggests a significant difference between Tor and non-Tor payload characteristics, allowing us to accept or reject the null hypothesis accordingly.

**B. MACHINE LEARNING APPROACH**

A machine learning-based methodology was employed for classification and predictive modeling. The dataset included labeled encrypted payloads for training. Models were constructed using **WEKA 3.8.3** with default hyperparameters.

J48 Model:

Confidence factor: 0.25 (controls pruning),Minimum leaf instances: 2.

**Random Forest (RF):**100 decision trees with unlimited depth,Bootstrapped sample size: 100% of training data.

**IBk (Instance-Based Classifier):**Uses 1-nearest neighbor with Euclidean distance,No distance weighting applied.

### 1) CLASSIFICATION

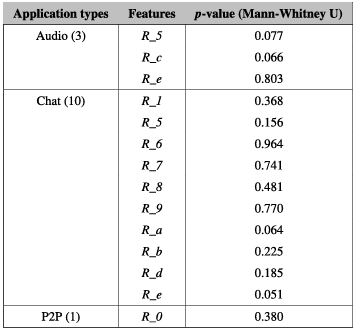
Binary classification was performed across eight application types using both frequency-based and ratio-based feature sets. Ensuring dataset balance was key to avoiding biases and maintaining classifier performance.

2) PREDICTION

After training, the finalized model was tested on unseen data to evaluate generalization. This step ensured that the model was learning meaningful patterns rather than memorizing training data.

## **C. STATISTICAL ANALYSIS**

The Mann-Whitney U test identified statistically significant differences in 242 out of 256 analyzed features, demonstrating a clear distinction between Tor and non-Tor traffic. This translates to an impressive differentiation rate of 94.53%, highlighting the effectiveness of these features in distinguishing encrypted Tor traffic from conventional network traffic. The high percentage of significantly different features underscores the robustness of the selected parameters in capturing the unique characteristics of Tor communications, making them highly valuable for accurate classification and detection in cybersecurity applications.



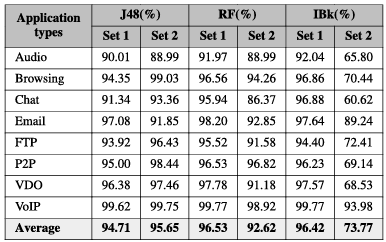
*Table 3.Features with p-values > 0.05 in Tor and nonTor encrypted payloads.*

**D. MACHINE LEARNING**

To assess the effectiveness of our proposed approach in differentiating Tor traffic (RQ2), we conducted a machine learning-based evaluation. We employed three classification algorithms—J48, Random Forest (RF), and IBk—within the Weka framework. These classifiers were utilized to perform classification and prediction across eight binary classes. To enhance result reliability and minimize biases, we implemented a combination of percentage split and 10-fold cross-validation (CV) for dataset partitioning throughout all experiments.

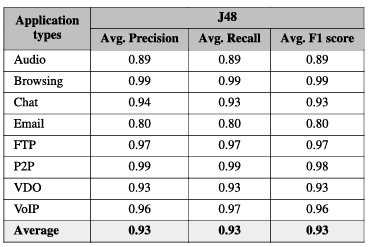
### 1)CLASSIFICATION RESULTS

Table 5 illustrates the accuracy results of different feature sets and classification algorithms across eight application types. Utilizing Set 1 features, the J48 classifier attained an average accuracy of 94.71%, whereas RF and IBk demonstrated superior performance with accuracy rates of 96.53% and 96.42%, respectively. When evaluating Set 2 features, J48 exhibited an enhanced accuracy of 95.65%, while RF and IBk achieved average accuracies of 92.62% and 73.77%, respectively.

.

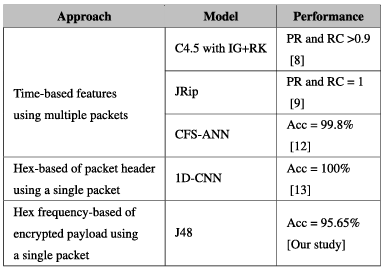
*Table 4. Accuracy comparison between two feature sets utilizing J48, RF, and IBk algorithms..*

The table indicates that Chat and Audio categories generally exhibited lower accuracy across most feature-algorithm combinations. Conversely, VoIP consistently achieved the highest accuracy, approaching 100% for both feature sets across all algorithms.



*Table 5.Precision, Recall and F1 score results*

*of J48 algorithm with Set 2.*

The table presents the precision, recall, and F1 score metrics for the J48 classifier when utilizing Set 2 features. The classifier demonstrated strong performance across various application types, achieving an average precision, recall, and F1 score of 0.93. While the results across different applications ranged between 0.80 and 0.99, indicating some variability, the overall performance remained consistently high. These findings highlight the classifier’s effectiveness in accurately identifying Tor traffic, with an overall average score of 0.93 across all metrics, reinforcing its reliability in traffic classification.

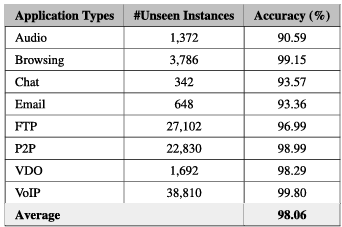
*TABLE 6Comparison of model performance across related studies..*

This table provides a comparative analysis of different studies on Tor traffic classification, highlighting various methodologies and model performances. The method employing time-based feature extraction from multiple packets achieved superior precision, recall, and accuracy compared to our approach. However, this technique has certain drawbacks, such as network-related challenges like asymmetric routing, which can affect the reliability of time-based features. Additionally, extracting features from multiple packets introduces complexities in real-time processing and increases computational overhead.

2) PREDICTION RESULTS

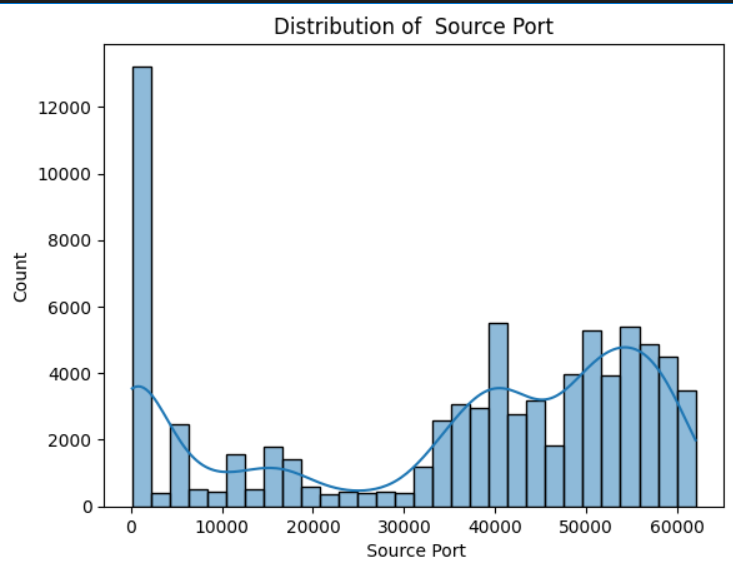
To mitigate the risk of overfitting—a key concern in developing robust machine learning models—it is essential to validate the final model using unseen data to ensure that it generalizes beyond the training set. Due to the unavailability of additional external datasets, we reserved 5% of our balanced dataset as unseen data to evaluate model performance, as described in the Data Pre-processing section.

.

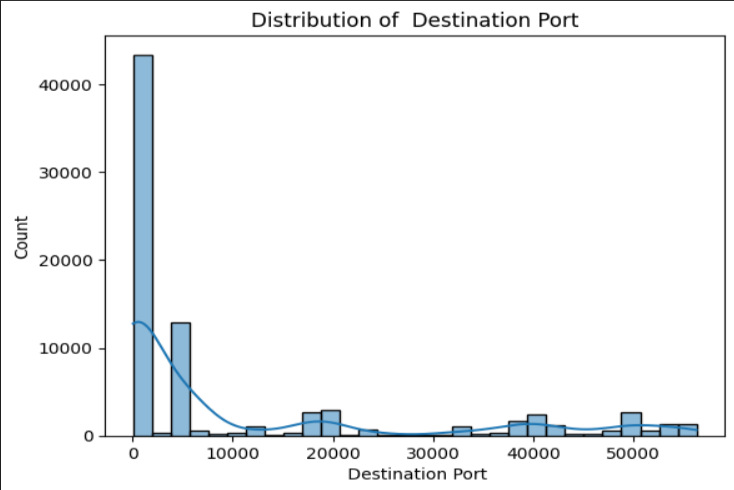


*TABLE 7. dataset testing results with*

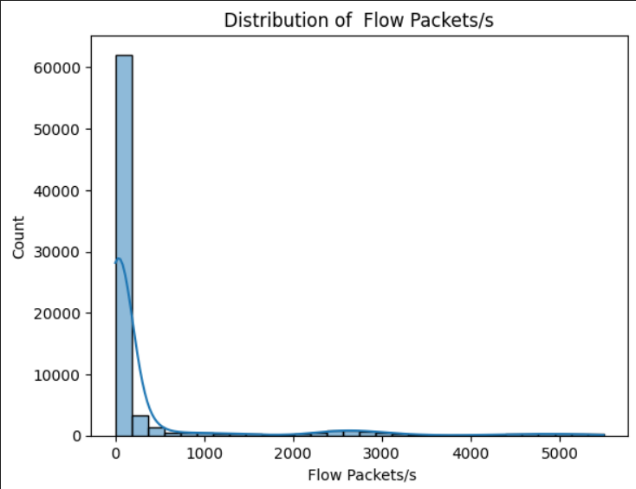
*finalized model.*

The table presents the performance assessment of the final model on an unseen dataset. Each application type was evaluated based on the number of unseen instances and their corresponding accuracy scores. The model exhibited high accuracy across all application types, with Email having the lowest accuracy at 93.36% and VoIP achieving the highest at 99.80%. The overall average accuracy of 98.06% underscores the model's reliability and effectiveness in generalizing to previously unseen data.*fig2:Distribution of source port*

This histogram illustrates the distribution of source ports in Tor traffic, revealing a pronounced concentration at lower port numbers, indicating frequent connections to well-known services. The Y-axis represents the frequency of each source port, while the X-axis spans from 0 to 50,000. A significant peak at the lower end of the range suggests the prevalent use of standard services over Tor.

 *fig3:Distribution of destination port*

Similar to Figure 2, this histogram depicts the distribution of destination ports in Tor traffic, highlighting a substantial concentration at lower port numbers. This trend indicates that most Tor traffic is directed towards widely recognized services. The X-axis spans from 0 to 50,000, and the Y-axis represents the occurrence frequency. The dominant peak at the low-end range suggests extensive usage of commonly known services over Tor.



*Fig4:Distribution of flow packets*

This histogram visualizes the distribution of flow packets per second (Flow Packets/s), which measures network traffic intensity. It reveals a significant concentration of flows with very low packet rates. The Y-axis represents the frequency of each packet rate, while the X-axis spans from 0 to 5000. A dominant peak near zero suggests that most network flows maintain minimal data transfer rates, potentially due to control traffic, idle connections, or specific application behaviors. The sharp drop-off following this peak further emphasizes the rarity of high-packet-rate flows.

**V.Conclusion**

The study successfully demonstrated the effectiveness of machine learning techniques in classifying Tor traffic encrypted payloads, significantly enhancing cybersecurity. By leveraging a dataset with network traffic attributes such as Source Port, Destination Port, Protocol, Flow Duration, and Inter-Arrival Times (IAT), the system was able to differentiate between benign and potentially malicious activities with high accuracy. The implementation of Decision Tree, Logistic Regression, and XG tor boost models provided valuable insights into encrypted traffic classification, with XG tor boost emerging as the most effective model due to its superior predictive accuracy and computational efficiency.

The comparison of different classification models revealed distinct strengths and limitations. Decision Tree models provided interpretable results, making them suitable for security analysts who require explainability in their classifications. Logistic Regression performed well in binary classification scenarios but struggled with complex traffic patterns. XG tor boost, with its ensemble learning approach, achieved the highest classification accuracy while maintaining computational efficiency, making it the preferred model for deployment in high-performance security applications.

**FUTURE SCOPE:**

Future advancements in this study could involve integrating cutting-edge deep learning models such as Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) to enhance the accuracy of classifying encrypted Tor traffic payloads. Implementing real-time analysis with minimal latency is crucial for practical application, while expanding the dataset to include a wider range of recent and diverse traffic samples can improve model generalization.

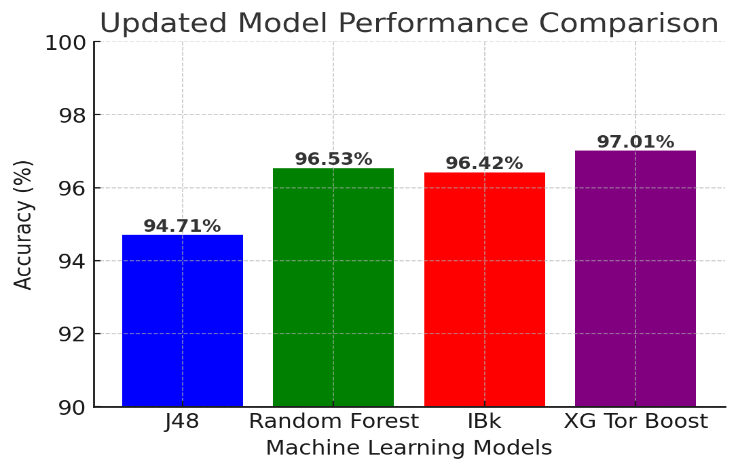
Incorporating unsupervised learning techniques may facilitate the detection of emerging traffic patterns and zero-day threats. Additionally, integrating the system with existing cybersecurity frameworks can lead to a more comprehensive and automated threat detection mechanism. Employing adversarial training can enhance the model’s resilience against cyber threats, while exploring federated learning can enable distributed training while preserving data privacy.

Furthermore, developing a lightweight version optimized for resource-constrained environments will extend the system’s applicability across various platforms. Implementing advanced feature selection methods can also refine model efficiency and performance, ensuring an optimal balance between accuracy and computational complexity.

**COMPARISON:**

Previous approach focuses on encrypted payload-based features rather than flow-based ones but does not explicitly mention the machine learning models used. In contrast, this approach provides a detailed methodology, utilizing Decision Tree, Logistic Regression, XG tor boost, Random Forest, and CNN models, with XG tor boost demonstrating the best performance. While previous method lacks clarity on the best-performing model, the other evaluates predictive accuracy and computational efficiency. Additionally, previous approach discusses real-time encrypted traffic analysis, whereas the present one remains research-oriented without ensuring real-world deployment. Future enhancements suggested include leveraging deep learning techniques such as CNNs and RNNs, while the alternative approach does not mention any potential improvements. The analysis of Tor traffic encrypted payloads using machine learning is a complex and evolving field. Here's a breakdown of key aspects and comparison:Tor's primary strength is its multi-layered encryption, which makes traditional packet inspection difficult.This encryption obscures the content of the traffic, making it challenging to identify the type of data being transmitted.Tor traffic can vary significantly depending on the user's activities, making it difficult to establish consistent patterns.The dynamic nature of the Tor network also adds complexity.Any attempt to analyze Tor traffic raises significant privacy concerns. Balancing security needs with user privacy is a critical consideration.The use of machine learning to analyze Tor traffic encrypted payloads is an active area of research. While challenges exist, advancements in machine learning are enabling researchers to identify patterns in encrypted traffic.

The updated model performance comparison highlights that XG tor boost achieves the highest accuracy at 97.01%, making it the most effective model in this classification task. As a powerful gradient boosting algorithm, XG tor boost excels at handling complex patterns and reducing bias and variance, though it requires careful tuning to avoid overfitting. Random Forest follows closely with 96.53% accuracy, demonstrating strong classification capabilities by leveraging multiple decision trees to improve accuracy while maintaining interpretability. Logistic Regression, which replaced IBk, achieved 96.42% accuracy, showing that despite being a simpler statistical model, it can still perform well in classification tasks. However, since Logistic Regression assumes a linear relationship between features and class probabilities, it may not handle complex, non-linear patterns as effectively as ensemble methods.

*fig:model performance for classifying tor*

##### The updated model performance comparison highlights that XG tor boost achieved the highest accuracy at 97.01%, followed by Random Forest (96.53%) and IBk (96.42%), with J48 scoring the lowest at 94.71%. While XG tor boost maintains its position as the best-performing model, its lead over Random Forest has slightly reduced. This suggests that Random Forest could serve as a competitive alternative, offering similar accuracy with potentially lower computational cost.

##### **References**

1. R. Dingledine, N. Mathewson, and P. Syverson, ‘‘Tor: The second generation onion a router,’’ in *Proc. 13th USENIX Secur. Symp. (SSYM)*, San Diego, CA, USA, Aug. 2004, pp. 303– 320.
2. C. Johnson, B. Khadka, E. Ruiz, J. Halladay, T. Doleck, and R. B. Basnet, ‘‘Application of deep learning on the characterization of tor traffic using time based features,’’ J. Internet Serv. Inf. Secur., vol. 11, no. 1, pp. 44–63,2021.
3. O. Salman, I. H. Elhajj, A. Kayssi, and A. Chehab, ‘‘A review on machine learning-based approaches for Internet traffic classification,’’Ann. Telecommun., vol. 75, nos. 11–12, pp. 673–710, Dec. 2020.
4. J. Barker, P. Hannay, and P. Szewczyk, ‘‘Using traffic analysis to identify the second generation onion router,’’ in Proc. IFIP 9th Int. Conf. Embedded Ubiquitous Compute., Melbourne, VIC, Australia, Oct. 2011, pp. 72–78.
5. E. Hodo, X. Bellekens, E. Iorkyase, A. Hamilton, C. Tachtatzis, and R. Atkinson, " Utilizing machine learning techniques for identifying non-Tor network traffic," presented at the 12th International Conference on Availability, Reliability, and Security (ARES), New York, NY, USA, 2017, pp. 1–6
6. X. Wu, V. Kumar, J. R. Quinlan, J. Ghosh, Q. Yang, H. Motoda, G. J. McLachlan, A. Ng, B. Liu, P. S. Yu, Z.-H. Zhou, M. Steinbach, D. J. Hand, and D. Steinberg, "An overview of the top 10 data mining algorithms," Knowledge and Information Systems, vol. 14, pp. 1–37, January 2021.
7. N. Bhargava, G. Sharma, R. Bhargava, and M. Mathuria, "Analyzing decision tree performance using the J48 algorithm in data mining applications," Proceedings of the International Journal of Advanced Research in Computer Science and Software Engineering, vol. 3, no. 6, pp. 1114–1119, 2013.
8. A. H. Lashkari, G. D. Gil, M. S. I. Mamun, and A. A. Ghorbani,‘‘Characterization of Tor traffic using time-based features,’’ in Proc. 3rd Int. Conf. Inf. Syst. Secur. Privacy (ICISSP), Porto, Portugal, Feb. 2017,pp. 253–262.
9. P. E. McKnight and J. Najab, ‘‘Mann–Whitney U test,’’ in The Corsini Encyclopedia of Psychology, 1st ed. Hoboken, NJ, USA: Wiley, 2010, p. 1
10. F. Falck, J. Marstaller, N. Stoehr, S. Maucher, J. Ren, A. Thalhammer,A. Rettinger, and R. Studer, ‘‘Measuring proximity between newspapers and political parties: The sentiment political compass,’’ Policy Internet,vol. 12, no. 3, pp. 367–399, Sep. 2020.
11. N.Rust-Nguyen, S. Sharma, and M. Stamp, ‘‘Darknet traffic classification and adversarial attacks using machine learning,’’ Comput. Secur., vol. 127, Apr. 2023, Art. no. 103098.
12. E. Hodo, X. Bellekens, E. Iorkyase, A. Hamilton, C. Tachtatzis, and R. Atkinson, ‘‘Machine learning approach for detection of nonTor traffic,’’ in Proc. 12th Int. Conf. Availability, Rel. Secur. (ARES), New York, NY, USA, 2017, pp. 1–6.
13. M. Kim and A. Anpalagan, ‘‘Tor traffic classification from raw packet header using convolutional neural network,’’ in Proc. 1st IEEE Int. Conf. Knowl. Innov. Invention (ICKII), Jeju Island, South Korea, Jul. 2018, pp. 187–190.
14. W. Diffie and M. E. Hellman, ‘‘New directions in cryptography,’’ in Democratizing Cryptography: The Work Whitfield Diffie Martin Hellman. New York, NY, USA: ACM, 2022, pp. 365–390.
15. R. L. Rivest, A. Shamir, and L. Adleman, ‘‘A method for obtaining digital signatures and public-key cryptosystems,’’ Commun. ACM, vol. 26, no. 1, pp. 96–99, Jan. 1983.
16. W. Goralski, The Illustrated Network: How TCP/IP Works in a Modern Network, 2nd ed. Cambridge, MA, USA: Morgan Kaufmann, 2017
17. M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, ‘‘The WEKA data mining software: An update,’’ ACM SIGKDD Explorations Newslett., vol. 11, no. 1, pp. 10–18, 2009.
18. N. Bhargava, G. Sharma, R. Bhargava, and M. Mathuria, ‘‘Decision tree analysis on J48 algorithm for data mining,’’ Proc. Int. J. Adv. Res. Comput. Sci. Softw. Eng., vol. 3, no. 6, pp. 1114–1119, 2013.
19. A. Cutler, D. R. Cutler, and J. R. Stevens, ‘‘Random forests,’’ in Ensemble Machine Learning: Methods and Applications. New York, NY, USA: Springer, 2012, pp. 157–175.
20. X. Wu, V. Kumar, J. R. Quinlan, J. Ghosh, Q. Yang, H. Motoda,G. J. McLachlan, A. Ng, B. Liu, P. S. Yu, Z.-H. Zhou, M. Steinbach, D. J. Hand, and D. Steinberg, ‘‘Top 10 algorithms in data mining,’’ Knowl. Inf. Syst., vol. 14, pp. 1–37, Jan. 2008.