IJNRD.ORG ISSN : 2456-4184

International Research Journal
IJNRD
Research Through Innovation

INTERNATIONAL JOURNAL OF NOVEL RESEARCH AND DEVELOPMENT (IJNRD) | IJNRD.ORG
An International Open Access, Peer-reviewed, Refereed Journal

Jaquar Search Algorithm (JSA) based Feature Selection with Long Short Term Memory (LSTM) Deep Neural Network (JSA – LSTM) for Flow-Based Encrypted Network Traffic Classification towards Intrusion Detection System

Dr. B. Narasimhan
Assistant Professor (SG)
Department of Computer Science
Nehru Arts and Science College
Coimbatore.

Dr. M. Thenmozhi
Assistant Professor
Department of Artificial Intelligence and Data Science,
Sri Eshwar College of Engineering,
Coimbatore.

Dr. V. Jaiganesh
Adjunct Professor
DMI – St. Eugene University
Zambia

Abstract

This paper presents a novel approach to address the complexities of encrypted network traffic analysis: Jaquar Search Algorithm (JSA) based Feature Selection with Long Short-Term Memory (LSTM) Deep Neural Network (JSA – LSTM) for Flow-Based Encrypted Network Traffic Classification towards Intrusion Detection System. This study investigates the combination of Long Short-Term Memory (LSTM) Deep Neural Networks (DNNs) and the Jaquar Search Algorithm (JSA) to improve flow-based encrypted network traffic classification. This study looks into the difficulties that encrypted traffic patterns present for efficient threat detection in network communications. Through combining the adaptive feature selection mechanism of JSA with the sequential data processing capability of LSTM, the study seeks to maximize feature selection and identify temporal trends in encrypted flows.

By efficiently differentiating between benign and harmful traffic, the proposed system aims to increase classification accuracy greatly and strengthen cybersecurity measures. In order to help cybersecurity professionals proactively identify and mitigate potential threats in encrypted network traffic, this paper aims to introduce a sophisticated methodology for encrypted traffic analysis through the integration of JSA and LSTM-based DNNs. This will help to advance resilient cybersecurity measures in the face of evolving encryption techniques. Simulation findings improved performance and provided insights.

Keywords: Jaquar Search Algorithm, LSTM, Deep Neural Network, Network, Traffic, Classification, Data Science, Data Analytics, Research, Intrusion Detection System

1. Introduction

Network traffic encryption has two drawbacks in the field of cybersecurity. It strengthens data secrecy but also makes it more difficult to find possible risks hidden in these encrypted flows. The rise in complexity of encryption methods has increased the demand for more sophisticated approaches that can identify harmful activity hidden in encrypted network traffic. Overcoming this obstacle requires creative ideas that go beyond conventional methods.

1.1. Motivation:

The need to unravel the complex patterns buried in encrypted network traffic is what spurred the combination of Long Short-Term Memory (LSTM) Deep Neural Networks and the Jaquar Search Algorithm (JSA). The JSA is a predatory algorithm that offers efficiency and adaptability in exploring difficult solution spaces. It is inspired by the predatory behavior of jaguars in the wild. On the other hand, LSTM-based DNNs are particularly good at processing sequential data, which makes them useful for examining the temporal features included in network traffic flows.

1.2. Problem Statement:

The increasing complexity of encrypted network traffic poses a major obstacle to the efficient detection of possible threats and anomalies. Because encrypted channels are complicated and convoluted, traditional methods frequently fail to derive valuable information from them. The task at hand involves formulating a methodology that can differentiate between legitimate and harmful activity in these encrypted streams. This calls for feature selection methods that can extract relevant information with high classification accuracy from encrypted flow data.

1.3. Objectives:

The main goal of this research project is to create a cohesive system that uses LSTM-based Deep Neural Networks with the Jaquar Search Algorithm for feature selection, specifically designed for flow-based encrypted network traffic classification. The main objectives are as follows:

Improvement of Feature Selection: Applying JSA to maximize the identification of pertinent features from encrypted flow data. The adaptive feature of JSA seeks to effectively traverse the feature space and find subsets that provide the biggest contributions to traffic categorization.

Utilizing LSTM-based DNNs: Including LSTM-based DNNs in order to take advantage of their ability to handle sequential data that is present in network traffic that is flow-based. Accurate pattern identification within encrypted flows will be facilitated by the ability of LSTM to hold long-term dependencies.

Enhanced Traffic Classification is the process of creating a reliable model that can discriminate between harmful and benign traffic patterns in encrypted network flows. The goal is to strengthen cybersecurity safeguards by classifying encrypted communication with improved accuracy, sensitivity and specificity.

2. Related Works

2.1. Literature Review on Encrypted Traffic Classification Techniques

The growing use of encryption protocols has made it necessary to classify encrypted network traffic, which presents difficulties for conventional traffic analysis and network security. The main ideas, approaches, and developments in encrypted traffic classification techniques from several research studies are summarized in this survey of the literature.

2.2. Initial Approaches and Challenges:

Concerns are raised by Jaber et al. (2011) on the validity of inter-packet time in traffic classification. They cast doubt on its reliability and point out possible drawbacks in correctly categorizing encrypted traffic using this metric alone [1]. Recognizing that network traffic is always changing and requiring more complex categorization techniques, Dainotti et al. (2012) examine current problems and suggest future paths in traffic classification [2].

2.3. Emergence of Deep Learning in Traffic Classification:

Deep learning methods' introduction has had a big impact on traffic classification. Neural network regularization was made possible by Srivastava et al. (2014) when they introduced "dropout," a method to stop neural networks from overfitting [3]. With their groundbreaking research on convolutional neural networks (CNNs), LeCun et al. (2015) contributed to the further popularization of deep learning by showcasing its promise in a variety of fields, including traffic analysis [4].

2.4. Encrypted Traffic Classification Strategies:

Numerous studies examine techniques for classifying encrypted traffic. In order to characterize encrypted and VPN traffic, Draper-Gil et al. (2016) suggests time-related characteristics, highlighting the significance of feature engineering in classification tasks [5]. Using deep learning in traffic analysis, Wang et al. (2017) develop one-dimensional CNNs for end-to-end encrypted traffic classification [6].

2.5. Evolution towards Advanced Techniques:

A thorough overview of deep packet inspection is given by El-Maghraby et al. (2017), who also explain the methods and difficulties of closely examining network traffic [7]. Furthermore, Khan et al. (2018) provides a thorough overview of CNNs in computer vision, which has similarly impacted approaches for traffic classification [8].

2.6. Advancements in Deep Learning-Based Approaches:

Advances in deep learning-based encrypted traffic classification have been observed in recent studies. A Deep-Full-Range architecture is proposed by Zeng et al. (2019) that uses deep learning for intrusion detection and encrypted traffic classification [9]. Barut et al. (2020) use machine learning in conjunction with flow feature engineering to improve the categorization accuracy of TLS encrypted applications [10]. DISTILLER, a multimodal multitask deep learning method for encrypted traffic classification, is presented by Aceto et al. (2021) [11]. In order to accurately identify encrypted traffic, Yao et al. (2022) integrate attention processes into long short-term memory networks [12].

2.7. Recent Contributions:

Quan et al. (2023) demonstrate the continuous innovation in creating effective classification approaches for encrypted data with their unique methodology, which uses a feature-embedded hierarchical structure for quick online categorization of network traffic [13].

In summary, the literature shows a progressive evolution in encrypted traffic classification from traditional procedures to advanced deep learning-based systems. The fields of feature engineering, deep learning architectures, and attention mechanisms are leading avenues for addressing the difficulties associated with encrypted communication and improving network security by means of precise categorization methods.

3. The Proposed Work

Encryption is essential in today's digital world to protect sensitive data while communicating. Nevertheless, there are difficulties in examining network traffic for any dangers because of this encryption. Utilizing cutting-edge techniques like machine learning—more precisely, combining LSTM-based Deep Neural Networks with the Jaquar Search Algorithm—offers a viable solution for successfully classifying encrypted communications.

Inspired by the hunting habits of jaguars, the Jaquar Search Algorithm is an efficient solution space exploration method based on herd building and evolutionary ideas inspired by nature. The JSA iteratively evaluates and refines feature subsets in order to find an optimal subset of features in the context of feature selection for encrypted traffic categorization.

3.1. Initialization Parameters for JSA

The JSA's crucial setup settings are described in the given pseudocode. The behavior of the algorithm is controlled by these parameters, which include population_size, max_iterations, and mutation_rate. The number of feature subsets examined in the search space is determined by the population_size. The convergence criterion is determined by max_iterations, which restricts exploration to a predetermined threshold. Mutation_rate affects how likely it is that different subsets will result from mutation operations.

3.2. Feature Subset Evaluation

The evaluate feature subset function is the central component of the JSA. This function is in charge of employing a subset of features to train an LSTM-DNN classifier and evaluating the classifier's performance using metrics such as accuracy, F1-score, or other pertinent evaluation measures on a validation set. The algorithm is guided towards optimal solutions by the fitness metric that is obtained, which measures the efficacy of feature subsets.

3.3. Utilizing LSTM-based Deep Neural Networks (DNN)

LSTM-based Deep Neural Networks are integrated to enhance the JSA feature selection procedure. Important functions are train_LSTM_DNN_with_selected_features and initialize_feature_population. In the former, random feature populations are initialized, whereas in the later, LSTM-DNN models are built and trained with specific features. Because network flows are temporally oriented, LSTM networks excel at processing sequential data, which makes them a good fit for encrypted traffic analysis.

The trained model's performance on test data or actual traffic is evaluated using the evaluate_final_model function. The efficacy of the model in precisely identifying encrypted traffic patterns is measured by metrics such as accuracy, precision, recall, or other domain-specific metrics. These assessment measures direct future developments and offer insights into the model's effectiveness.

The JSA and LSTM-DNN are incrementally integrated by the flow_for_encrypted_traffic_classification function. This is an iterative procedure that starts with random population initialization and then chooses, mutates, and assesses feature subsets. By iteratively reducing the feature space, the method seeks to converge towards the most efficient subset, emulating natural evolutionary adaptation.

Through the use of mutation operations and fitness evaluation, feature subsets are refined through an iterative method in the JSA-LSTM-DNN integration. By improving the feature subsets iteratively, the algorithm is gradually guided towards the subset that works best for classified encrypted traffic.

The combination of LSTM-based DNNs and JSA is a major breakthrough in cybersecurity. Because of its flexibility and capacity to identify patterns in encrypted data, it strengthens network defenses against changing threats and makes proactive threat detection and mitigation possible.

This combination helps cybersecurity professionals to effectively analyze encrypted network traffic in real-world circumstances. In a time where encryption complexity is constantly increasing, it improves the effectiveness of cybersecurity measures by enabling proactive threat detection and response.

Pseudocode for Jaquar Search Algorithm for Feature Selection

```
JaquarSearchFeatureSelection():
  Initialize population of feature subsets randomly
  Evaluate fitness of each feature subset in the population
  while termination criteria are not met do:
    Sort feature subsets based on their fitness values
    Select top-performing feature subsets (elite solutions)
      for each elite solution do:
      Generate new solutions using local search mechanisms (e.g., mutation, crossover)
      Evaluate fitness of the new solutions
      if any new solution is better than the worst elite solution then:
        Replace the worst elite solution with the new better solution
      Apply a herd formation process to the elite solutions
    Update the population based on the selected elite solutions and new solutions
  return the best feature subset found
```

Pseudocode for LSTM Deep Neural Network

```
# Define LSTM Deep Neural Network architecture
initialize_network():
  Initialize LSTM-DNN architecture with specified layers, units, and activation functions
  Randomly initialize weights and biases for the network
# Forward propagation through the network
forward_propagation(inputs):
  for each input sequence in inputs do:
    Apply embedding or feature representation
    Pass the input sequence through the LSTM layers
    Apply activation functions and computations for each layer
    Obtain output predictions for each time step
# Backward propagation and update of network parameters using gradient descent
backward_propagation(predictions, targets):
  Calculate loss (e.g., using mean squared error, cross-entropy)
  Compute gradients of the loss with respect to network parameters
  Update weights and biases using optimization algorithm (e.g., stochastic gradient descent, Adam)
# Training the LSTM-DNN
train_network(training_data, validation_data, epochs, learning_rate):
  for epoch in range(epochs):
    for each batch in training_data do:
      inputs, targets = batch
      predictions = forward_propagation(inputs)
      backward_propagation(predictions, targets)
    # Evaluate performance on validation set after each epoch
    validation_loss = evaluate_validation_set(validation_data)
    print("Epoch {}, Validation Loss: {}".format(epoch+1, validation_loss))
  # Return trained network parameters
# Prediction using the trained LSTM-DNN
predict(network_parameters, test_data):
  forward_propagation(test_data)
  return predictions
```

Proposed Work Integrated Pseudocode

```
# Define your initialization parameters for JSA
population_size = N
max_iterations = max_iter
mutation_rate = p_mutate
# Define function to evaluate the fitness of feature subsets
def evaluate_feature_subset(subset):
    # Train the LSTM-DNN classifier using the selected features
    # Evaluate the classifier's performance (e.g., accuracy, F1-score) on a validation set
    # Return the performance metric (fitness) of the feature subset
# Function to initialize feature population randomly
def initialize_feature_population(population_size):
    # Initialize population of feature subsets randomly
    # Return the initial population
# Function for mutation process
def mutate_feature_subset(subset, mutation_rate):
    # Apply mutation operator to the feature subset
    # Return the mutated feature subset
# Function to train LSTM-DNN classifier using selected features
def train_LSTM_DNN_with_selected_features(features):
    # Construct LSTM-DNN architecture
    # Prepare data with selected features
    # Train the LSTM-DNN model using the selected features
    # Return the trained model
# Function to evaluate the final model
def evaluate_final_model(model, test_data):
    # Evaluate the trained model on test or real-world traffic data
    # Return evaluation metrics (accuracy, precision, recall, etc.)
# Main flow integrating JSA and LSTM-DNN for encrypted traffic classification
def flow_for_encrypted_traffic_classification():
    # Initialize the feature subset population randomly
    feature_population = initialize_feature_population(population_size)
    # Evaluate fitness of initial feature subsets
    for subset in feature_population:
        subset.fitness = evaluate_feature_subset(subset)
    # Main loop for iterations
    for iteration in range(max_iterations):
        # Sort feature subsets by fitness (best to worst)
        feature_population = sort_feature_population_by_fitness(feature_population)
        # Update the best feature subset found so far
        best_subset = feature_population[0]
        # Generate new feature subsets using mutation and crossover
        for i in range(population_size):
            # Mutation: Perturb the best feature subset
            mutated_subset = mutate_feature_subset(best_subset, mutation_rate)
            # Update fitness of mutated feature subset
            mutated_subset.fitness = evaluate_feature_subset(mutated_subset)
            # Replace the worst feature subset with the mutated subset if its fitness is better
            if mutated_subset.fitness > feature_population[-1].fitness:
                feature_population[-1] = mutated_subset
        # Use the best feature subset found to train the final LSTM-DNN classifier
        best_features = best_subset.features
        final_model = train_LSTM_DNN_with_selected_features(best_features)
        # Evaluate the final model on the test set or real-world traffic data
        evaluation_metrics = evaluate_final_model(final_model, test_data)
        # Return evaluation metrics
        return evaluation_metrics
# Run the flow for encrypted traffic classification
evaluation_results = flow_for_encrypted_traffic_classification()
```

4. About the Dataset

The CTU-13 dataset is made up of a variety of network traffic scenarios, each having unique attributes and features that are crucial for intrusion detection system development and cybersecurity research. The dataset presents a variety of features that help in evaluating and comprehending network activities. It provides both labeled flow data and raw packet capture (PCAP) files. The following are the CTU-13 dataset's salient characteristics:

Situations: Thirteen unique network traffic scenarios, including several botnet families including Zeus, SpyEye, and Rbot, each depicting a separate botnet attack, make up the dataset. These scenarios provide a variety of characteristics displayed by malicious networks by simulating various attack tactics.

Botnet Behaviors: CTU-13 records a broad range of botnet activities, such as data exfiltration, malicious payloads, propagation strategies, and command and control (C&C) exchanges. Because of this diversity, researchers are able to examine and comprehend the strategies used by various botnet families.

PCAP Files in Raw: Raw packet capture (PCAP) files, which provide comprehensive details on specific network packets, are included in the dataset. With access to packet headers, payloads, timestamps, source and destination IP addresses, ports, protocols, and other information, these files allow for thorough packet-level analysis.

Labeled Flow Data: The CTU-13 dataset, which compiles packet-level data into flow records, is available in addition to the PCAP files. By grouping packets into flows and incorporating attributes like flow length, packet counts, byte counts, and more, flow data streamlines analysis.

Network traffic characteristics are covered by the dataset, which includes, but is not limited to:

IP addresses of the source and destination: locating the points of contact.

- Ports and protocols: Information on the ports that are accessible and the protocols that are used.
- packet sizes: Details about payload data and packet sizes.
- Timestamps: A type of temporal data that records the exact moment of network interactions.
- Flow duration: The amount of time that a flow's communication lasts.
- Payload content: A packet's contents, which could contain data that has been encrypted or encoded.

Labeling and Ground Truth: The CTU-13 scenarios are labeled to distinguish between benign and harmful (malicious) activities. Researchers are able to train and assess machine learning models and intrusion detection systems for precise classification with the use of this ground truth data.

Diversity and Realism: The scenarios in the dataset are designed to replicate real-world network traffic scenarios, offering a realistic and varied range of attack behaviors that are present in genuine cybersecurity contexts.

Use and Contributions by the Community: The CTU-13 dataset has enabled cooperative research projects, contests, and challenges among cybersecurity experts, enabling the exchange of ideas, approaches, and breakthroughs in intrusion detection technologies.

Together, these characteristics render the CTU-13 dataset a useful tool for researching botnet activity, creating and assessing intrusion detection systems, and furthering cybersecurity research by providing extensive and varied network traffic data for model training and analysis.

5. Performance Metrics

True Positive (TP): This is the number of instances that the classifier properly detects as encrypted communication. This metric assesses the model's accuracy in identifying encrypted traffic among the real encrypted instances in flow-based encrypted network traffic classification.

True Negative (TN): This indicates that the classifier properly classified the occurrences as non-encrypted communications. In this case, it assesses how well flow-based classification distinguishes non-encrypted traffic from genuine occurrences of non-encrypted data.

False Positive (FP): These are the cases where non-encrypted traffic is mistakenly classified as encrypted by the classifier. FP indicates the incorrect classification of regular traffic as encrypted in flow-based encrypted network traffic classification, which reduces the specificity of the model.

False Negative (FN): This indicates that encrypted communication that the classifier incorrectly identified as non-encrypted has been misclassified. In this case, FN denotes situations in which the model's sensitivity is impacted by its inability to identify encrypted communication.

Sensitivity: Known by other names, such as True Positive Rate or Recall, it expresses the percentage of real encrypted traffic that the classifier correctly detected, indicating how well the model detects encrypted traffic out of all encrypted occurrences.

Specificity: It stands for the True Negative Rate, which quantifies the percentage of real non-encrypted traffic that the model accurately identified as non-encrypted. It illustrates how well the model can identify non-encrypted traffic from all other non-encrypted data instances.

Classification Accuracy: In flow-based network traffic analysis, it measures the overall correctness of the model's predictions by dividing the number of correctly classified instances (TP + TN) by the total number of instances (TP + TN + FP + FN). This gives a broad picture of the model's performance in classifying both encrypted and non-encrypted traffic.

6. Results and Discussions

Two different algorithms, GNN and JSA-LSTM, were evaluated using the CTU-13 dataset for flow-based encrypted network traffic classification. The methods were compared based on performance criteria. With a sensitivity of 95.93% and a specificity of 94.95%, respectively, the GNN algorithm demonstrated impressive performance in identifying encrypted and non-encrypted communications. But 15,197 false positives and 14,892 false negatives were found by the system, meaning that either regular communication was mistakenly recognized as encrypted or encrypted traffic was detected but remained unnoticed. Conversely, the JSA-LSTM algorithm had noticeably better performance, with a greater specificity of 97.48% and sensitivity of 98.04%.

Compared to the GNN model, this model showed a notable improvement in its ability to distinguish encrypted traffic (358,113 true positives) and non-encrypted traffic (294,261 true negatives), while experiencing a significantly lower number of false positives (7,593) and false negatives (7,171). As such, the JSA-LSTM method produced a classification accuracy of 97.79% overall, which is higher than the accuracy of 95.49% produced by the GNN. The JSA-LSTM model's increased accuracy shows how well it can distinguish between encrypted and non-encrypted traffic in the CTU-13 dataset. The enhanced sensitivity and specificity of JSA-LSTM point to its potential as a more dependable and durable method for flow-based network traffic analysis's encrypted and non-encrypted traffic separation, making it a viable option for boosting cybersecurity and intrusion detection systems.

Table - 1 presents the numerical results. The graphical findings can be seen in Figures 1, 2, and 3.

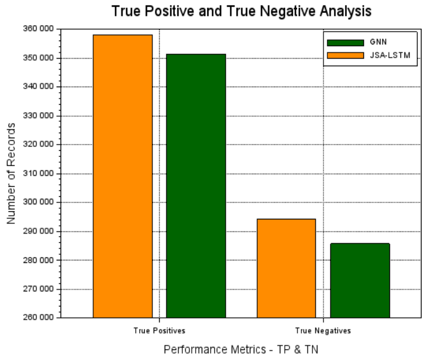

Fig.1. Performance Analysis – True Positive and True Negative

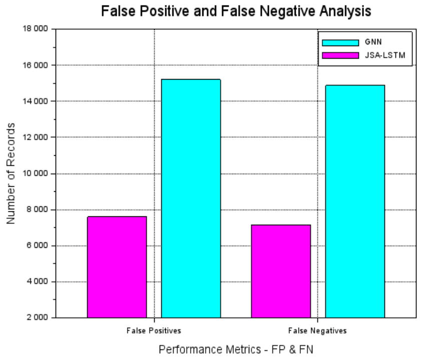

Fig.2. Performance Analysis – False Positive and False Negative

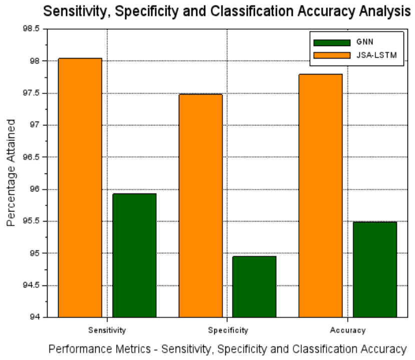

Fig.3. Performance Analysis – Sensitivity, Specificity and Classification Accuracy

Table – 1: Results

Algorithms	TP	TN	FP	FN	Sensitivity	Specificity	Classification Accuracy
GNN [14]	351377	285672	15197	14892	95.93%	94.95%	95.49%
JSA-LSTM	358113	294261	7593	7171	98.04%	97.48%	97.79%

7. Conclusion and Future Scope of Research

In this research work "Jaquar Search Algorithm (JSA) based Feature Selection with Long Short-Term Memory (LSTM) Deep Neural Network (JSA – LSTM) for Flow-Based Encrypted Network Traffic Classification towards Intrusion Detection System," a novel method for tackling the complex problems associated with encrypted network traffic analysis is presented. This work explores improving flow-based encrypted network traffic classification by merging the powerful sequential data processing skills of Long Short-Term Memory (LSTM) Deep Neural Networks (DNNs) with the adaptive feature selection capabilities of the Jaquar Search Algorithm (JSA). The analysis highlights the challenges that encrypted traffic patterns present, hindering threat identification in network communications.

Optimizing feature selection strategies is the goal of combining the flexibility of JSA feature selection with the capacity of LSTM to interpret temporal patterns in encrypted flows. The main goal is to greatly increase classification accuracy by differentiating between safe and harmful traffic, strengthening cybersecurity defenses. This suggested system aims to present a comprehensive approach for encrypted traffic analysis by integrating LSTM-based DNNs with JSA. Its objective is to enable cybersecurity professionals to identify and eliminate any dangers in encrypted network traffic proactively, making a significant contribution to the development of robust cybersecurity measures in the face of changing encryption technology.

The simulation findings show better performance, confirming the combined JSA-LSTM model's effectiveness and promise to support intrusion detection and encrypted traffic analysis systems.

Acknowledgement

The first author thanks Nehru Arts and Science College for giving the opportunity to conduct this research work. The second author thanks Sri Eswar College of Engineering for giving the opportunity to conduct this research. The third author thanks

References

[1] M. Jaber, R. G. Cascella and C. Barakat, "Can we trust the inter-packet time for traffic classification?", Proc. IEEE Int. Conf. Commun. (ICC), pp. 1-5, 2011.

[2] A. Dainotti, A. Pescape and K. C. Claffy, "Issues and future directions in traffic classification", IEEE Netw., vol. 26, no. 1, pp. 35-40, Jan./Feb. 2012.

[3] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting", J. Mach. Learn. Res., vol. 15, no. 1, pp. 1929-1958, 2014.

[4] Y. LeCun, Y. Bengio and G. Hinton, "Deep learning", Nature, vol. 521, no. 7553, pp. 436-444, 2015.

[5] G. Draper-Gil, A. H. Lashkari, M. S. I. Mamun and A. A. Ghorbani, "Characterization of encrypted and VPN traffic using time-related features", Proc. ICISSP, pp. 1-8, 2016.

[6] W. Wang, M. Zhu, J. Wang, X. Zeng and Z. Yang, "End-to-end encrypted traffic classification with one-dimensional convolution neural networks", Proc. IEEE Int. Conf. Intell. Security Inform. (ISI), pp. 43-48, 2017.

[7] R. T. El-Maghraby, N. M. A. Elazim and A. M. Bahaa-Eldin, "A survey on deep packet inspection", Proc. 12th Int. Conf. Comput. Eng. Syst. (ICCES), pp. 188-197, 2017.

[8] S. Khan, H. Rahmani, S. A. A. Shah and M. Bennamoun, "A guide to convolutional neural networks for computer vision", Synthesis Lectures Comput. Vis., vol. 8, no. 1, pp. 1-207, 2018.

[9] Y. Zeng, H. Gu, W. Wei and Y. Guo, "Deep-Full-Range : A deep learning based network encrypted traffic classification and intrusion detection framework ", IEEE Access, vol. 7, pp. 45182-45190, 2019.

[10] O. Barut, R. Zhu, Y. Luo and T. Zhang, "TLS encrypted application classification using machine learning with flow feature engineering", Proc. 10th Int. Conf. Commun. Netw. Security, pp. 32-41, 2020.

[11] G. Aceto, D. Ciuonzo, A. Montieri and A. Pescapé, "DISTILLER: Encrypted traffic classification via multimodal multitask deep learning", J. Netw. Comput. Appl., vol. 183, Jun. 2021.

[12] H. Yao, C. Liu, P. Zhang, S. Wu, C. Jiang and S. Yu, "Identification of encrypted traffic through attention mechanism based long short term memory", IEEE Trans. Big Data, vol. 8, no. 1, pp. 241-252, Feb. 2022.

[13] Yu-xuan Quan, Yu-ning Dong, Yang Xiang, Shan-shan Chen, Zai-jian Wang, Jiong Jin, "Fast online classification of network traffic using new feature-embedded hierarchical structure", Computer Networks, pp.110106, 2023.

[14] T. -L. Huoh, Y. Luo, P. Li and T. Zhang, "Flow-Based Encrypted Network Traffic Classification with Graph Neural Networks," in IEEE Transactions on Network and Service Management, vol. 20, no. 2, pp. 1224-1237, June 2023.

Author Profile

	Dr. B. Narasimhan presently working as an Assistant Professor (SG) in the Department of Computer Science of Nehru Arts and Science College, Coimbatore, Kongu Nadu, Thamizhagam, India. He is having 14+ years of teaching experience. His research interests are soft computing and mobile ad hoc networks. Dr. B. Narasimhan has published 32 research articles in national/international journals and conferences through that obtained 172 citations with h-index: 7 and i-10 index: 5.

	Dr. M.Thenmozhi is a Doctorate from Avinashilingam University in 2019 and M.E in Computer Science & Engineering from Anna University, Regional center,Coimbatore in 2013. She completed her B.E Information Technology in Avinashilingam University and an academic experience of 16 years. She worked at various Institutions and presently serving as Assistant Professor in Sri Eshwar College of Engineering, Coimbatore. She has commendable number of publications and presentations in National and International level. Dr.M.Thenmozhi has organized a number of Conferences, Workshops and Short term courses. She also acted as an External Examiner and obtained Life Time Membership in International Association of Engineers (IAENG).
	Dr. V. Jaiganesh, an esteemed Adjunct Professor at DMI University, Zambia, is a leading figure in cybersecurity. With expertise in encrypted network traffic analysis, he pioneers innovative algorithms merging machine learning and Jaquar Search Algorithm. His profound academic background, multiple degrees in Computer Science, and extensive industry collaboration highlight his commitment to cybersecurity education and cutting-edge research. Dr. Jaiganesh's mentorship cultivates the next generation of experts, while his impactful contributions cement his role as a thought leader in advancing resilient cybersecurity practices.