Ranah Research
Journal of Multidisciplinary Research and Development
E-ISSN: 2655-0865
DINASTI RESEARCH
082170743613 ranahresearch@gmail.com https://jurnal.ranahresearch.com

DOI: https://doi.org/10.38035/rrj.v8i2

Combining Random Forest with Firefly Algorithm to Improve Darknet Traffic Detection

Vincent Timothy Lim[1], Rusdianto Roestam[2]
[1]Master of Science in Information Technology, President University, Cikarang, Indonesia, vincent.lim@student.president.ac.id
[2]Master of Science in Information Technology, President University, Cikarang, Indonesia, rusdianto@president.ac.id

Corresponding Author: vincent.lim@student.president.ac.id[1]

Abstract: Darknet traffic detection system, a cyber-crime activity detection system that detects the use of Tor and VPNs, is one way to reduce the occurrence of darknet cyber-crimes. Current existing detection tools such as machine learning models have shown its capability in detecting darknet network traffics. However, it still faces some limitations in its performance due to suboptimal hyperparameters. One of the existing classification models used for darknet traffic detection, such as Random Forest demonstrated great performance in detecting darknet activities. This research utilizes the Firefly Algorithm (FA), a prominent swarm intelligence method, to fine-tune hyperparameters and enhance the detection capabilities of the Random Forest (RF) model. The proposed RF-FA (Random Forest – Firefly Algorithm) approach is evaluated against the standard Random Forest model. Tests performed on the CIC-Darknet2020 dataset reveal that the Firefly Algorithm improves the RF model's performance in all key metrics. The optimized RF-FA model attains an accuracy, precision, recall, and F1-score of 98.73%, surpassing the baseline RF model, which achieves 98.62% in accuracy, precision, and recall, along with an F1-score of 98.61%.

Keyword: Darknet Traffic, Tor (The Onion Router), VPN (Virtual Private Network), Random Forest, Firefly Algorithm.

INTRODUCTION

Following the onset of the COVID-19 pandemic, darknet usage has escalated, with a marked increase in involvement from amateur cybercriminals (Almomani, 2023). The darknet enables users to distribute and access information, including illicit content related to drug trafficking, arms sales, and cybercrime (Karunanayake et al., 2023). This underscores the critical need for an effective detection system to monitor and identify illegal activities within darknet traffic.

Despite progress in machine and deep learning for darknet traffic classification, model accuracy can still be enhanced. Previous work (Iliadis & Kaifas, 2021) confirmed Random

Forest's performance with default hyperparameters, but research on optimizing it using the Firefly Algorithm—especially for darknet traffic—remains limited (Kumar & Kumar, 2020).

Existing work (Iliadis & Kaifas, 2021) has established that machine learning methods can successfully classify darknet traffic. But their performance is hindered by suboptimal hyperparameter settings, which reduce their accuracy. Manual hyperparameter tuning can be time-consuming and does not guarantee optimal result (Sarwar et al., 2021). Proper hyperparameter adjustment is critical for maximizing model accuracy and computational efficiency in machine learning.

The objective of this work is to advance the precision of machine learning-based darknet traffic classification, specifically for Tor and VPN services. Through quantitative evaluation of standard performance indicators and a comparative analysis against the existing Random Forest benchmark (Iliadis & Kaifas, 2021), the study aims to demonstrate measurable enhancements. Ultimately, the study aims to demonstrate that hyperparameter tuning is essential for improving model performance.

LITERATURE REVIEW

Prior studies (Iliadis & Kaifas, 2021) have explored multiple machine learning techniques for darknet traffic classification, such as k-Nearest Neighbors (kNN), Multi-layer Perceptron (MLP), Random Forest, Decision Tree, and Gradient Boost algorithms. The result shows the capability of Random Forest with an accuracy of 98.62%. The training process utilizes the CIC-Darknet2020 dataset, a standardized benchmark specifically developed for darknet traffic analysis applications.

Additional studies, including (Almomani, 2023), have validated the effectiveness of Support Vector Machines (SVM), Convolutional Neural Networks (CNN), and the discriminator model from Auxiliary Classifier Generative Adversarial Networks (AC-GAN) as classification methods.

Prior research (Mane et al., 2019), (Saleem et al., 2024) has demonstrated the significant efficacy of machine learning models in network traffic classification. These approaches facilitate automated threat detection while enhancing both the speed and precision of cybersecurity systems. Through analysis of extensive network traffic datasets, machine learning algorithms can effectively differentiate between benign and malicious activities by identifying underlying data patterns.

Research (Aliefa & Suyanto, 2020) effectively employed a variable-length chromosome genetic algorithm (VLC-GA) to enhance recurrent neural network (RNN) architecture optimization, demonstrating notable improvements in computational efficiency for scaled network implementations. The study further demonstrated that VLC-GA outperformed reinforcement learning-based neural architecture search (NAS) in language modeling applications, achieving superior perplexity scores.

A key consideration in classification models is overfitting. Random Forest addresses this limitation of Decision Trees by generating multiple trees during training and aggregating their predictions through majority voting. This reduces the risk of overfitting and increases accuracy in classification (Iliadis & Kaifas, 2021), (Mane et al., 2019), (Speiser et al., 2019), (Coutinho Marim et al., 2023) Swarm intelligence-based algorithms can also solved overfitting problem as in (Clarissa & Suyanto, 2019), (Chioran & Valean, 2020), (Tawakkal & Suyanto, 2020). Another significant challenge in network traffic classification is the rise of encrypted communication, such as VPNs (Virtual Private Networks) and other encrypted protocols, which make it difficult to inspect the content of the traffic. However, machine learning models can classify encrypted traffic by detecting patterns in flow behavior (Draper-Gil et al., 2016).

Random Forest

As an ensemble learning method, Random Forest (RF) has gained widespread use in classification and regression problems. The algorithm generates an ensemble of decision trees, with each tree trained on a bootstrap sample of the data and restricted to random feature subsets for node splitting, effectively minimizing overfitting potential (Speiser et al., 2019).

A recent advancement in Random Forest methodology is the development of the Linear Random Forest (LRF) algorithm. The LRF modifies the traditional decision trees in the forest by using linear decision boundaries rather than axis-aligned splits (Ao et al., 2019). This method effectively models datasets with strong linear feature relationships—a capability lacking in conventional decision tree approaches.

Random Forest's key advantage is its strong generalization capability, enabling robust performance on previously unseen data. By constructing each decision tree on a random feature subset, the algorithm prevents excessive specialization to the training data. Furthermore, the ensemble approach of combining multiple trees reduces prediction variance without increasing bias, establishing Random Forest as a highly versatile and robust machine learning method.

Firefly Algorithm

The Firefly Algorithm (FA) is a nature-inspired metaheuristic optimization technique that mimics the photic communication patterns of fireflies through bioluminescence. In this approach, each solution is represented as a firefly whose luminance corresponds to solution quality, with less optimal solutions naturally gravitating toward better ones. This biological analogy enables effective navigation through high-dimensional search spaces for complex optimization challenges.

In machine learning, finding the right hyperparameters is essential to achieving optimal performance. Suboptimal hyperparameter configurations may lead to either underfitting or overfitting, ultimately compromising the model's generalization performance on unseen data. Hyperparameter tuning can improve accuracy, robustness, and the overall performance of a model (Feurer & Hutter, 2019). The performance of Random Forest models is heavily influenced by key hyperparameters including n_estimators (tree count), max_depth (tree depth limit), and max_features (feature subset size per split), as demonstrated in (Probst et al., 2019).

The Firefly Algorithm offers several advantages over traditional optimization algorithms. A primary advantage of the Firefly Algorithm is its capacity to escape local optima—unlike gradient-based methods such as Gradient Descent, which often converge prematurely. FA's randomness and firefly movement based on brightness allow it to explore the search space more thoroughly, making it suitable for complex optimization tasks where local optima can be problematic (Bernal et al., 2021).

According to Xin-She Yang on (Yang, 2014), the Firefly Algorithm (FA) involves several formulas and concepts.

The Euclidean distance r_{ij} between fireflies i and j is computed as:

$$r_{ij} = ||x_i - x_j|| = \sqrt{\sum_{k=1}^{d} (x_{i,k} - x_{j,k})^2}$$

where $x_{i,k}$ denotes the k-th coordinate of firefly i's position in d-dimensional space.
The attractiveness β between fireflies follows an exponential decay model:

$$\beta(r) = \beta_0 e^{-\gamma r^2}$$

where:

- β_0 represents the baseline attractiveness at zero distance.
- γ denotes the light absorption coefficient.
- r is the Euclidean separation between fireflies.

The position update for firefly i moving toward brighter firefly j follows:

$$x_i = x_i + \beta_0 e^{-\gamma r_{ij}^2}(x_j - x_i) + \alpha(\text{rand} - 0.5)$$

where:

- x_i, x_j denote current positions
- $\beta_0 e^{-\gamma r_{ij}^2}$ is the distance-dependent attractiveness
- α controls random exploration
- $rand \sim$ Uniform [0, 1]

METHOD

Dataset

This study employs the CIC-Darknet2020 benchmark dataset, a purpose-built collection specifically designed for darknet traffic analysis and classification tasks. The dataset comprises 85 features with over 141,000 rows of data that represent different aspects of network traffic, such as flow characteristics, packet lengths, timestamps, inter-arrival times, and bytes transmitted. Here are some of the features in the dataset.

Tabel 1. Cic-Darknet2020 Dataset Feature Samples

Feature	Description
Flow ID	Unique identifier for each flow of network traffic
Source IP	Originating host's IP address
Source Port	Communication endpoint on source host
Destination IP	Target host's IP address
Destination Port	Target service port number
Protocol	Transport layer protocol identifier (TCP/UDP/etc.)
Timestamp	Time at which the flow occurred
Flow Duration	Temporal length of the communication session (ms/s)
Forward Packets	Count of source→destination packets
Backward Packets	Count of destination→source packets
Byte Rate	Throughput in bytes per second (Bps)
Packet Rate	Transmission frequency (pps)
Label	Classification label (e.g., Tor, Non-Tor, VPN, Non-VPN)
Detailed Traffic Label	Further classification (e.g., Audio Streaming, Browsing)

Data Pre-processing

Data preprocessing is a critical stage in machine learning pipelines, involving data cleaning, transformation, and normalization to ensure optimal model performance and reliable analytical results.

Data Cleaning is the first pre-processing step in this research. Missing values and inconsistent entries (such as NaN values or infinite values) are identified and removed. Missing or corrupted data instances are addressed through either listwise deletion or statistical imputation methods, with the approach selected based on data characteristics and analysis requirements.

The next step is Label Encoding, where the categorical labels (e.g., Tor, Non-Tor, VPN) are converted into numerical format using label encoding. This is necessary for the machine

learning model to process these labels during the training phase. Each category is assigned a unique numerical value to represent the type of traffic.

After that, Feature Scaling is conducted to ensure that all features contribute equally to the model, continuous features such as packet lengths, flow duration, and byte counts are normalized or scaled. This helps avoid dominance of large-value features over smaller ones and improves the convergence of the model training process. The process will use the $StandardScaler$ formula for each numerical column.

$$x' = \frac{x - \mu}{\sigma}$$

The dataset is partitioned using an 80-20 stratified split, with 80% allocated for model training and the remaining 20% reserved as a holdout test set to evaluate generalization capability on unseen samples.

Hyperparameter Tuning and Model Training

Hyperparameter tuning aims to identify the optimal configuration of model parameters that maximize predictive performance on unseen data. In this research, the hyperparameter tuning is conducted using a Firefly Algorithm, for a Random Forest model.

The optimization process initiates by creating a population of candidate solutions, with each firefly representing a distinct combination of Random Forest hyperparameters. These parameters include:

1. $n\_estimators$: Number of trees in the Random Forest, randomly initialized within [50, 200].
2. max_depth: Maximum depth of the trees, randomly initialized within [5, 30].
3. $min\_samples\_split$: Minimum samples required to split a node, within [2, 10].
4. $min\_samples\_leaf$: Minimum samples required to form a leaf node, within [1, 5].

During initialization, each firefly's luminance is quantified by: (1) configuring a Random Forest model with its hyperparameter set, (2) training on the designated subset, and (3) computing the weighted F1-score from classification metrics on the test data, which directly corresponds to the firefly's brightness value.

A firefly with lower brightness will move towards a brighter one using the movement formula. After each movement, hyperparameters are clipped to ensure they remain within valid bounds. To maintain diversity in the population and avoid premature convergence, a mutation step is included with a 10% probability. This step replaces a firefly's hyperparameters with new random values.

The algorithm iteratively performs three core operations - brightness evaluation, positional updates, and mutation - across N generations. Each generation's optimal solution is determined by selecting the firefly with maximum luminance (highest F1-score). For this research, a number of 10 generations and 10 population size is used.

The optimized hyperparameters are utilized to initialize the Random Forest model, which is subsequently trained on the feature matrix X_train and target vector y_train using a supervised learning paradigm.

Evaluation

Performance evaluation is critical for validating the model's classification efficacy on darknet traffic. The study utilizes a suite of standard classification metrics—including accuracy, precision, recall, F1-score, and AUC-ROC—to holistically evaluate the optimized Random Forest model's detection performance across all critical dimensions. The evaluation metrics and their respective formula are as follow.

Accuracy represents the proportion of correctly classified instances among all predictions, combining both true positives (correct threat detections) and true negatives (proper benign traffic identification).

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Precision represents the ratio of correctly identified positive cases (true positives) to all predicted positive cases (true positives + false positives). For darknet classification, this metric indicates how trustworthy positive detections are.

$$Precision = \frac{TP}{TP + FP}$$

The recall metric evaluates the model's ability to identify all positive instances, calculated as the ratio of correctly predicted positives to all actual positives. In darknet detection, this reflects the system's coverage of malicious traffic.

$$Recall = \frac{TP}{TP + FN}$$

The F1-Score represents the harmonic mean of precision and recall, providing a balanced evaluation metric. In darknet detection contexts, an elevated F1-Score signifies optimal trade-off between minimizing false alarms (precision) and maximizing threat detection (recall).

$$F1\ Score = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

RESULTS AND DISCUSSION

Table 2 presents a comparative performance analysis between the baseline Random Forest (RF) and its Firefly Algorithm-optimized counterpart (RF-FA) on the CIC-Darknet2020 dataset. The baseline performance metrics for the original Random Forest model were obtained from prior research (Iliadis & Kaifas, 2021), which established benchmark results using identical evaluation protocols on the same dataset. Both of the model is trained using the same CIC-Darknet2020 dataset to classify between 4 classification labels (Tor, Non-Tor, VPN, Non-VPN).

Tabel 2. Evaluation Metrics Of Original Rf And Rf-Fa Model

Metric	Original RF	RF-FA	Improvement (%)
Accuracy	98,62%	98,73%	+0,11%
Precision	0,9862	0.987325	+0,001125
Recall	0,9862	0.987278	+0,001078
F1-score	0,9861	0.987263	+0,001163

The proposed RF-FA performs much better overall than the original RF model. The optimized RF-FA model achieves superior performance, attaining 98.73% classification accuracy and an F1-score of 0.9873, while the baseline RF demonstrates marginally lower metrics (accuracy: 98.62%; F1-score: 0.9861). The model achieves 98.73% accuracy, indicating that it correctly classifies 98.73% of all instances (both malicious and benign traffic) in the test set. This demonstrates strong overall discriminative capability in darknet traffic classification.

The RF-FA model demonstrates a precision of 0.9873 (98.73%), indicating that 98.73% of instances classified as positive (darknet traffic) were correctly identified, while only 1.27% represented false positives. This suggests that when the model predicts a positive outcome,

there is a 98.73% chance that the prediction is correct, which is slightly higher compared to the RF model.

The RF-FA model achieves a recall (sensitivity) of 0.9873 (98.73%), demonstrating its capacity to correctly identify 98.73% of actual darknet traffic instances while failing to detect only 1.27% of positive cases.

CONCLUSION

The proposed model of RF-FA is successfully developed to detect darknet traffics. A reliable comparison between RF model and the RF-FA model is also successfully conducted. The comparative performance analysis substantiates the experimental findings, confirming that the Firefly Algorithm effectively enhances Random Forest's detection capabilities for darknet traffic classification tasks. Since the CIC-Darknet2020 dataset consist of network traffic samples, this model is able to be applied in real-world implementation. Application and implementation to a server or a router for cyber security system can be subject as future works.

REFERENCE

Aliefa, M. H., & Suyanto. (2020). Variable-length chromosome for optimizing the structure of recurrent neural network. In *2020 International Conference on Data Science and Its Applications (ICoDSA)* (pp. 1–5). IEEE. https://doi.org/10.1109/ICoDSA50139.2020.9213012

Almomani, A. (2023). Darknet traffic analysis and classification system based on modified stacking ensemble learning algorithms. *Information Systems and E-Business Management*.

Almomani, A. (2023). Darknet traffic classification and adversarial system based on modified stacking ensemble learning algorithms. *Information Systems and E-Business Management*.

Ao, Y., Li, H., Zhu, L., Ali, S., & Yang, Z. (2019). The linear random forest algorithm and its advantages in machine learning assisted logging regression modeling. *Journal of Petroleum Science and Engineering, 174*, 776–789. https://doi.org/10.1016/j.petrol.2018.11.067

Bernal, E., Lagunes, M. L., Castillo, O., Soria, J., & Valdez, F. (2021). Optimization of type-2 fuzzy logic controller design using the GSO and FA algorithms. *International Journal of Fuzzy Systems, 23*(1), 42–57. https://doi.org/10.1007/s40815-020-00976-w

Chioran, D., & Valean, H. (2020). Arduino based smart home automation system. *International Journal of Advanced Computer Science and Applications, 11*(4). https://doi.org/10.14569/IJACSA.2020.0110410

Clarissa, V., & Suyanto. (2019). New reward-based movement to improve globally-evolved BCO in nurse rostering problem. In *2019 International Seminar on Research of Information Technology and Intelligent Systems (ISRITI)* (pp. 114–117). IEEE. https://doi.org/10.1109/ISRITI48646.2019.9034669

Coutinho Marim, M., et al. (2023). Darknet traffic detection and characterization with models based on decision trees and neural networks. *Intelligent Systems with Applications, 18*, 200199. https://doi.org/10.1016/j.iswa.2023.200199

Draper-Gil, G., Lashkari, A. H., Mamun, M. S. I., & Ghorbani, A. A. (2016). Characterization of encrypted and VPN traffic using time-related features. In *Proceedings of the 2nd International Conference on Information Systems Security and Privacy*.

Feurer, M., & Hutter, F. (2019). Hyperparameter optimization: Foundations, algorithms, best practices, and open challenges. *IFAC-PapersOnLine, 50*(1), 4973–4978. https://doi.org/10.1016/j.ifacol.2017.08.763

Iliadis, L. A., & Kaifas, T. (2021). Darknet traffic classification using machine learning techniques. In *2021 10th International Conference on Modern Circuits and Systems Technologies*.

Karunanayake, I., Ahmed, N., Malaney, R., Islam, R., & Jha, S. K. (2023). Darknet traffic analysis: Investigating the impact of modified Tor traffic on onion service traffic classification.

Kumar, V., & Kumar, D. (2020). A systematic review on firefly algorithm: Past, present, and future. *Archives of Computational Methods in Engineering*. https://doi.org/10.1007/s11831-020-09332-x

Mane, P., Sanghavi, V., Parkar, Y., Walanje, A., & Patel, J. (2019). Traffic classification using machine learning. In *Proceedings of the 2nd International Conference on Advances in Science & Technology (ICAST-2019)*.

Probst, P., Wright, M. N., & Boulesteix, A. (2019). Hyperparameters and tuning strategies for random forest. *WIREs Data Mining and Knowledge Discovery, 9*(3). https://doi.org/10.1002/widm.1301

Saleem, J., Islam, R., & Islam, M. Z. (2024). Darknet traffic analysis: A systematic literature review. *IEEE Access*.

Sarwar, M. B., Hanif, M. K., Talib, R., Younas, M., & Sarwar, M. U. (2021). DarkDetect: Darknet traffic detection and categorization using modified convolution-long short-term memory. *IEEE Access*.

Speiser, J. L., Miller, M. E., Tooze, J., & Ip, E. (2019). A comparison of random forest variable selection methods for classification prediction modeling. *Expert Systems with Applications, 134*, 93–101. https://doi.org/10.1016/j.eswa.2019.05.028

Tawakkal, M. I., & Suyanto. (2020). Exploration-exploitation balanced krill herd algorithm for thesis examination timetabling. In *2020 International Conference on Data Science and Its Applications (ICoDSA)* (pp. 1–5). IEEE. https://doi.org/10.1109/ICoDSA50139.2020.9212837

Yang, X.-S. (2014). Introduction to algorithms. In *Nature-inspired optimization algorithms* (pp. 1–21). Elsevier. https://doi.org/10.1016/B978-0-12-416743-8.00001-4