Network Requests Classification using Advanced Metaheuristic Optimization for Enhanced Network Security Systems

Marwa M. Eid[1,2,*]

[1]Faculty of Artificial Intelligence, Delta University for Science and Technology, Mansoura 11152, Egypt
[2]Jadara Research Center, Jadara University, Irbid 21110, Jordan

Email: mmm@ieee.org

Abstract

The importance of network security has greatly been enhanced in the modern digital environment that continuously changes. Network security, on the other hand, is a multi-layered defense mechanism that seeks to protect networks, data, and systems from malpractices such as unauthorized access breaches or activities. Cyber threats become ever more advanced, and traditional protective measures can no longer prove to be adequate. Given the necessity of such a threat to adapt and be intelligent, an active intrusion detection system must necessarily rapidly evolve its methods in response. The central element contained in this research is the proposal of a novel algorithm, BBERSC (Balance Between Al Biruni Earth Radius Optimization and Sine Cosine Algorithm). This algorithm is carefully crafted to achieve a compromise between the means for local search provided by Al-Biruni Earth Radius Optimization and probabilistic improvement, which are characteristic of the Swine Cosine Algorithm. BBERSC brings forward the cause of harmonizing these two optimization methods to revolutionize model accuracy and credibility, which may be achieved for network security's distinctiveness. One of the crucial elements of this study lies in the fact that hyperparameter tuning is quite a detailed process, especially for Random Forest. Parameters, including the number of trees, maximum depth, and minimum samples, are systematically employed to vary to augment pattern recognition capability by employing model processing network traffic. To ensure the validation of the effectiveness of the proposed models and algorithms, statistical analysis is carried out through ANOVA test & Wilcoxon Signed Rank Test. These tests show the models' results through rigorous assessments and emphasize differences between them. As the conclusion of this study, It is displayed that the Random Forest model utilized inside BBERSC algorithmic framework facilitates operational accuracy level 0.9901719, which is incomparable among all other machine learning algorithms.

Keywords: Network Requests; Al-Biruni Earth Radius Optimization; Sine Cosine Algorithm; Network Security; Intrusion Detection

1 Introduction

The environment of cyber systems has become a contemporary dynamic process where it is crucial to harden networks against an array of advanced threat actors that differ from one day to another [1]. What outweighs the simple act of introducing sturdy hardware and software solutions into the field is that this problem requires a

DOI: https://doi.org/10.54216/JAIM.100104

Received: March 28, 2025 Revised: June 18, 2025 Accepted: August 19, 2025

skilled arrangement of complex rules, configurations, and adaptive strategies to maneuver through an emerging threat terrain. The rapid progression of digital developments has spotlighted things even more, as it elevated the importance of preventing networks along with sensitizing what is needed in order to protect critical information from a widening arc of thoughtless complex breaches and ongoing attacks. To this complex and changing problem, the symbiotic coexistence of AI with optimization muses a powerful ally that offers optimized tools to strengthen the resilience and agility of digital infrastructure.

Today, meta-heuristic algorithms are used to solve a wide range of different problems because the optimization operation and search for the optimal solution from a large set of solutions is a necessary and vital thing in all models [2]. There are different search algorithms for optimization operations, which are exact, heuristic, and metaheuristic [3]. Exact algorithms are impractical in solving large problems because they often take a long time to solve large-scale problems [4]. Heuristic algorithms also mostly fall into the trap of local optimality and can no longer obtain high quality solutions in solving problems with high complexity [5]. However, meta-heuristic algorithms have a good ability to discover optimal or near-optimal solutions in an acceptable period of time [6], [7]. Meta-heuristic algorithms are a kind of heuristic algorithms that generate solutions by considering two components of intensification and diversity. In the diversity component, the focus is on producing diverse solutions in such a way that the entire problem space is explored [8]. In the intensification component, the focus is on searching the promising areas of the problem space. Therefore, the nature of the problem can show what kind of algorithm can get the results more accurately. But problems with discrete space require an algorithm with the ability to search in discrete space, which can be solved with genetic operators and neighborhood operators [9].

The rise of optimization, which can be seen in mechanics, economics, traveling planning, and internet routing, points out the continual existence of maximizing limited resources for a finite period with constrained funds as well as resources [10]. By adopting metaheuristic optimization algorithms following processes with advanced classification models, our research proposes to make network security stronger in this dynamic environment. Optimization serves as the basis for our factfinding, which is characterized by unique complexity associated with real-life paradoxes caused by nonlinearity and multimodality [11], [12].

The BBERSC optimization approach that is proposed seems to combine a nice balance of the Earth radius Al-Biruni (BER) and Sine Cosine Algorithm (CSA) with an ultimate result, which can reveal unique aspects of its workings supporting enhanced performance for classification models as shown in Figure 1. Starting from the collective intelligence concept and utilizing trigonometric functions, this approach makes our study results peculiar and workable through optimization methods.

From the unraveling of our exploration, an intricate network of optimization techniques converges collaboratively to sharpen the accuracy of the Random Forest Classifier. PSO and GWO may form a duo of swarm collective behavior-inspired methods, and WOA joins for support as a whale optimization algorithm. These algorithms go beyond a mere display of technical virtuosity to reflect the tenuous dance that is ubiquitous in biotypes and natural systems. This kind, as opposed to a narrow examination of possible answers, delves into potential solutions by taking an in-depth approach but is comprehensive nonetheless.

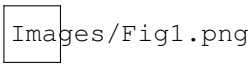

Figure 1: The balance between Al-Biruni earth radius optimization (BER) and Sine Cosine Algorithm (SCA) for Network Requests Dataset.

The capable classifiers within the arsenal of classification include an ensemble comprising Logistic Regression, Decision Tree, Random Forest [13], [14], support vector machine (SVM), k-nearest neighbors (KNN) [15], [16], Naive Bayes and SDG Classifier [17], [18] that can shed light on trends hidden in network requests for web and mobile apps [19]. We focus the next study on the critical evaluation of their performance [20],

DOI: https://doi.org/10.54216/JAIM.100104

Received: March 28, 2025 Revised: June 18, 2025 Accepted: August 19, 2025

using a wide range of metrics, including the above-listed dimensional attributes such as Accuracy, Sensitivity, Specificity, P-value, Positive Predictive Value (PPV), Negative Predictive Value (NPV), and F-Score.

The subsequent sections move smoothly into an intensive discussion of Related Works, Materials, and Methods as well as Experimental Results, enabling one to understand thoroughly all factors associated with network security optimization. These chapters systematize the process of thought, the carefully considered procedure, and the experimental results that define our story. The essence of the journey is crystallized in synthesized conclusions set out under the Conclusion and Future Directions sections, capturing as it does how AI models meshed with optimization entities offered a certain degree of protection against digital contortions.

The main contributions of this study include:

- Development of BBERSC Algorithm: This research presents a new optimization method, BBERSC (a Balance Between Al-Biruni Earth Radius Optimization and Sine Cosine Algorithm), that is being introduced. This algorithm wisely integrates the advantages of the Sine Cosine Algorithm with Al-Biruni Earth Radius Optimization. The goal is to maximize the efficiency of specific classification models designed specifically for network security functions and issues.
- Optimization for Network Security: Turning theory into practice goes beyond innovation but involves passing theoretical solutions from paperboard straight down to specific implementations that will be used, in this case, network security. The algorithm seeks to benefit from both two somewhat different optimization strategies by achieving a fine balance between them, thus increasing the classification of models, especially for identifying safe and unsafe requests in network traffic.
- Comprehensive Machine Learning Framework: The paper defines a complete machine learning infrastructure that combines leading optimization technologies with various basic and root-level models.
- Hyperparameter Tuning for Improved Random Forest: The study supports the hyperparameter tuning field because it involves the Random Forest model. The study seeks to attain optimal random forest classifier capabilities within network requests' classification by systematically investigating and fine-tuning vital hyperparameters, including tree number, maximum depths, and minimum samples.

These contributions together depict a well-balanced, integrative strategy for the problem of network security through optimization algorithms and machine learning models coupled with careful experimentation and comparison. Among many, the results are likely to provide useful findings for both intellectual and practical ends of cybersecurity research.

2 Related Works

The transformative paradigm shifts currently underway in the technological landscape, poised to redefine programming methodologies and alter interactions with the world as outlined in [21], is being discussed. Profound impacts have been exerted by prominent research domains, namely cloud computing and mobile computing, as well as the Internet of Things (IoT), which strives to establish an interconnected network of Internet-enabled devices for fostering intelligent environments. Within these domains and their intricate intersections, a plethora of emerging computing paradigms have surfaced, including Mobile Cloud Computing (MCC), cloudlet computing, mobile clouds, mobile IoT computing, IoT cloud computing, fog computing, Mobile Edge Computing (MEC), edge computing, the Web of Things (WoT), the Semantic WoT (SWoT), the Wisdom WoT (W2T), opportunistic sensing, participatory sensing, mobile crowdsensing, and mobile crowdsourcing. Regrettably, the lack of standardized definitions plagues these paradigms, leading to a confounding scenario where a single term might denote various paradigms, or conversely, several terms could

DOI: https://doi.org/10.54216/JAIM.100104

Received: March 28, 2025 Revised: June 18, 2025 Accepted: August 19, 2025

allude to a singular paradigm. Therefore, endeavors are being made to disentangle these paradigms, elucidating their positioning within the realms of cloud computing, mobile computing, and IoT. A historical trajectory, tracing these paradigms to their inception, is being meticulously pursued, accompanied by a comprehensive discourse on research trajectories within each domain. In the realm of the Internet of Things (IoT), analytics is emerging as an indispensable conduit for knowledge derivation and the facilitation of applications within smart homes, as discussed in [22]. The proliferation of connected appliances and devices within these homes engenders a substantial corpus of data, offering insights into consumer behavior and daily routines. Therefore, IoT analytics assumes a pivotal role in tailoring applications to meet the individual needs of homeowners while concurrently catering to the evolving requirements of burgeoning industries seeking to harness consumer profiles. The proposed framework advocates the integration of fog nodes and cloud systems to enable data-driven services, thereby surmounting the challenges posed by the intricacies and resource demands intrinsic to online and offline data processing, storage, and classification analysis. The requisite specifications and design components integral to the system are meticulously delineated. In an effort to validate the platform and furnish substantive outcomes, a case study is being presented utilizing a dataset sourced from an authentic smart home in Vancouver, Canada. The experimental results unequivocally underscore the efficacy and practical viability of the proposed platform.

Scientific workflows, characterized by a multitude of interdependent tasks, constitute a vital category within the domain of complex scientific applications, as discussed in [23]. Notably, a novel breed of serverless infrastructures has surfaced, exemplified by services like Google Cloud Functions and AWS Lambda, encapsulating the Function-as-a-Service model. These serverless infrastructures, designed initially to process background tasks in Web and Internet of Things applications or engage in event-driven stream processing, are being scrutinized. The investigation aims to assess their adaptability to more computationally and data-intensive scientific workflows, exploring potential avenues for repurposing serverless architectures in executing such workflows. Prototype workflow executor functions have been meticulously developed, leveraging AWS Lambda and Google Cloud Functions in tandem with the Hyper Flow workflow engine. These functions exhibit the capability to execute workflow tasks within AWS and Google infrastructures, featuring functionalities such as data staging to/from S3 or Google Cloud Storage and the execution of custom application binaries. The Montage astronomy workflow, a benchmark frequently employed in evaluations, has been successfully deployed and executed, with initial performance evaluation results detailed herein.

The findings underscore the user-friendly nature of this approach, albeit accompanied by costs associated with preparing portable application binaries for remote execution. While acknowledging the prototype status of the solution, this approach is perceived as highly promising. Future steps of the execution of scientific workflows in serverless infrastructures are being discussed alongside a comprehensive cost analysis and implications for resource management in scientific applications at large. The focal point of this study [24] is addressing the formidable challenge of delivering services for Internet of Things (IoT) applications with stringent real-time and predictable latency demands. IoT applications, relying on the interaction between devices and the physical environment, necessitate integration into Cloud and Fog computing paradigms due to the limited capabilities of IoT devices. Fog computing enhances service latency by bringing resources closer to the network edge. This study formulates IoT service request scheduling as an optimization problem, utilizing integer programming to minimize latency. Recognizing the NP-hard nature of the problem, a customized genetic algorithm (GA) is introduced as a heuristic approach to scheduling. In a simulation environment considering dynamic conditions, the GA outperforms other techniques, exhibiting an overall latency improvement of 21.9% to 46.6% and a notable enhancement in meeting request deadlines by up to 31%, demonstrating superior efficacy in optimizing IoT service request scheduling.

Extracting important features in network traffic to discover malicious agents using deep learning method can be a suitable method for intrusion detection. However, there are still many challenges in this field. Convolutional neural networks algorithm is a suitable method for feature extraction. In ‘Anomaly-based intrusion detection system in the Internet of Things using a convolutional neural network and multi-objective enhanced Capuchin Search Algorithm’ , CNN algorithm and Capuchin Search Algorithm (CSA) binary multi-objective optimization algorithm with feature selection approach for intrusion detection are proposed. This model has been tested and trained using TON-IoT and NSL-KDD datasets, and the results obtained from this model indicate that it performs well with an acceptable accuracy rate in detecting malicious agents. However, due to the use of a deep learning algorithm, this model can be very time-consuming. Often, to

DOI: https://doi.org/10.54216/JAIM.100104

Received: March 28, 2025 Revised: June 18, 2025 Accepted: August 19, 2025

increase the efficiency of an optimization algorithm in solving the problem of feature selection, they are combined with the use of improvement operators or combined with other optimization algorithms to reduce the weaknesses of an optimization algorithm. In 'An improved PIO feature selection algorithm for IoT network intrusion detection system based on ensemble learning ', the improved pigeon algorithm is combined with a local search algorithm to perform feature selection for intrusion detection. Also, an ensemble learning mechanism has been used to increase efficiency, and this approach is based on multiple one-class classifiers. To evaluate this model, different data sets called NLS-KDD, BoT-IoT, KDDCUPP99, and UNSW-NB15 have been used. This model has been compared with other algorithms using different criteria, and the results obtained from this comparison have shown that this model has an acceptable performance compared to the compared algorithms. As mentioned earlier, meta-heuristic algorithms perform well in feature selection. In another research,' Research on hybrid intrusion detection based on improved Harris Hawk optimization algorithm', an improved Harris Hawk optimization algorithm is proposed using a singer chaos map to select features and reduce noisy features to increase classifier performance. Chaos map, the singer map is used for initialization. Also, multi-information fusion has been used to select the best solution, and another different strategy has been used to obtain optimal features. The data is pre-processed using deep denoising autoencoder and k-nearest neighbor algorithms. Therefore, the imbalance between data in network traffic is eliminated. A deep neural network has been used to perform classification operations, and standard hobby datasets KDD CUP99, NSL-KDD, and UNSW-NB15 have been used to perform training and testing operations. The results shown from the tests performed in this test indicate the acceptable performance of this model in detecting penetration. However, this model also suffers from an almost long running time.

In the landscape of data processing, cloud computing assumes a pivotal role. However, the advent of the Internet of Things (IoT) brings about a surge in data generated from diverse devices, as discussed in [25]. Consequently, there arises a compelling necessity to align cloud characteristics with the request generator, allowing vast datasets to be processed in close proximity to the end user, just one hop away. Fog computing, a paradigm designed to furnish storage and computation capabilities at the network's edge, thereby mitigating network traffic and addressing inherent drawbacks of cloud computing, is given rise to. This paper delves into the taxonomy of fog computing, elucidating its distinctions from cloud computing and edge computing technologies. It expounds upon its applications, delving into the nuanced realm of emerging key technologies such as communication and storage technologies. Simultaneously, the paper navigates the intricate landscape of challenges inherent in fog technology, presenting a comprehensive exploration of this transformative paradigm. In the context of ITS, VANET is one must-have enabler that has drawn much attention among researchers and practitioners, as demonstrated in [26]. The rapid spread of vehicular applications, in addition to data explosion, has inevitably increased demand for communication, computing, and storage resources with strict requirements on response time and network capacity. Providing answers to these problems, MEC comes as a worthy answer that locally transfers computational and storage capabilities from the distant cloud into the network's edge closer to vehicular users. This movement allows for low latency and reduced bandwidth usage. The benefits of MEC have given rise to concerted efforts at vehicular networking and integration into MEC, marking a new revolutionary paradigm called Vehicular Edge Computing (VEC). In this paper, an in-depth review of the current state of VEC research is conducted. Starting with a broad general view covering introduction, architecture, prominent drivers, benefits, disadvantages, and promising real-world use cases of VEC, the paper develops into topics research in which VEC is researched.

3 Materials and Methods

The proposed methodology of classification of network requests using advanced Al-Biruni optimization algorithm, shown in Figure 2, is composed of the following steps:

1. Dataset Preprocessing:
    - Data Cleaning: Identify and handle missing dataset values. This step involves imputation techniques or removing instances with missing values.

DOI: https://doi.org/10.54216/JAIM.100104

Received: March 28, 2025 Revised: June 18, 2025 Accepted: August 19, 2025

- Feature Scaling: Normalize or standardize features to ensure that each feature contributes equally to the further processing.
- Encoding Categorical Variables: Convert categorical variables into numerical representations using techniques such as one-hot encoding or label encoding.
- Feature Selection: Select relevant features that contribute most to the classification task, potentially using techniques like feature importance or dimensionality reduction methods such as Principal Component Analysis.

2. Classification using Various Machine Learning Algorithms:

   - SGD Classifier: Stochastic Gradient Descent classifier that optimizes the loss function iteratively.
   - Logistic Regression: A linear model used for binary classification tasks.
   - Decision Tree Classifier: Constructs a decision tree based on feature values and predicts the class label for a given instance.
   - Gaussian Naive Bayes: Assumes that features are independent and follows a Gaussian distribution.
   - K-Nearest Neighbors (KNN): Classifies instances based on the majority class among their k nearest neighbors.
   - Support Vector Machines (SVM): Separates classes by finding the hyperplane that maximizes the margin between them.
   - Random Forest Classifier: Constructs an ensemble of decision trees and aggregates their predictions to improve accuracy and reduce overfitting.

3. Optimization of the Parameters of the RF Classifier using Advanced Biruni Optimization Algorithm:

   Advanced Biruni Optimization Algorithm is applied to fine-tune the hyperparameters of the Random Forest Classifier. This optimization involves optimizing parameters such as the number of trees, tree depth, and minimum samples per leaf node.

4. Classification using the Optimized RF:

   The Random Forest Classifier with optimized hyperparameters obtained from the Advanced Biruni Optimization Algorithm is applied to the dataset for classification. This step ensures that the classifier operates with the best possible configuration.

5. Measuring the Performance of the Optimized Classification based on the following criteria:

   - Accuracy: Calculated as the ratio of correctly classified instances to the total number of instances.
   - ROC Curve: Plots the true positive rate against the false positive rate across different threshold values.
   - Residual Analysis: Examines the differences between predicted and actual values to assess the model's performance.
   - Homoscedasticity: Tests whether the variance of the errors is constant across all levels of the predictor variables.
   - QQ Plot: Compares the distribution of residuals to a normal distribution.
   - Heatmap: Provides a visual representation of performance metrics across different scenarios, such as varying hyperparameters or feature combinations.
   - ANOVA and Wilcoxon Tests: Statistical tests to assess the significance of differences between groups or paired samples.
   - Regression Analysis: Examines the relationship between predictor variables and the target variable using regression techniques.

Figure 2: The steps of the proposed methodology.

DOI: https://doi.org/10.54216/JAIM.100104

Received: March 28, 2025 Revised: June 18, 2025 Accepted: August 19, 2025

3.1 Dataset

The data in this study was also drawn from Kaggle [27], a notable source of machine learning and data science sets. This dataset comprises a set of HTTP requests [28], where each request has parameters that capture the multiple associated aspects. Figure 3 represents the application of data as an illustration that is used to visualize the distribution of safe and unsafe requests for selected datasets. Such a graphical interpretation can be taken as an introduction to the description of characteristics that will set up the ground for this study and provide it with the necessary background.

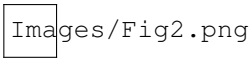

Figure 3: Safe and unsafe requests distribution in the selected dataset.

3.1.1 Proposed BBERSC algorithm

The BBERSC (Balance Between Al-Biruni Earth Radius Optimization and Sine Cosine Algorithm) algorithm is the basis of this study, which aims to achieve a balance between two algorithms – namely, the BER and the SCA. This algorithmic combination utilizes both optimization methods, highlighting each one's characteristics that serve as a stimulus, improving overall performance in the context of a given dataset. The BBERSC algorithm is shown in detail in Algorithm 1.

This algorithmic approach is essential for a successful, robust optimization process built specifically toward the dataset and complexity of the successive classification tasks. Its implementation and performance will be discussed in more detail along with the Experimental Results section, revealing its ability to increase the accuracy of the Random Forest Classifier.

3.2 Hyperparameters tuning

The search for an optimal model performance never ignores hyperparameter tuning just as much in Random Forest. The chosen hyperparameter set by Random Forest is a crucial determinant of the ability to spot patterns. We embark on tuning the following hyperparameters:

- n_estimators: Tree population in the forest. The higher number usually results in better performance, but it is associated with a significantly more significant amount of computational complexity. Finding the balance is key [29].
- max_features: The depth of the search tree. This can result in a higher value that produces more varied trees but also tends to raise problems of an over-fit. The most important thing is to determine the best number of feature subsets for each split [30].

DOI: https://doi.org/10.54216/JAIM.100104

Received: March 28, 2025 Revised: June 18, 2025 Accepted: August 19, 2025

Algorithm 1 : The proposed BBERSC optimization algorithm

Initialize BER population $\mathbf{P}_i(i = 1, 2, ..., d)$ with size d, iterations Max_{iter}, fitness function F_n, $t = 1$, n_1, n_2, a, r_1, r_2, r_3, r_4, r_5, r_6, r_7
Calculate fitness function F_n for each $\mathbf{P}_i$
Find best solution as $\mathbf{P}^*$
while $t \leq Max_{iter}$ **do**
  if $R \leq 0$ **then**
    for each solution in the exploration group **do**
      Heading towards the best solution
      $\mathbf{r} = h\frac{cos(x)}{1-cos(x)}$
      $\mathbf{D} = \mathbf{r}_1(\mathbf{P}(t) - 1)$
      $\mathbf{P}(t+1) = \mathbf{P}(t) + \mathbf{D}(2\mathbf{r}_2 - 1)$
    end for
    for each solution in the exploitation group **do**
      Elitism of the best solution
      $\mathbf{D} = \mathbf{r}_3(\mathbf{L}(t) - \mathbf{P}(t))$
      $\mathbf{P}(t+1) = \mathbf{r}_2(\mathbf{P}(t) + \mathbf{D})$
      $\mathbf{k} = 1 + \frac{2\times t^2}{Max_{iter}^2}$
      Investigate the area around best solutions as:
      $\mathbf{P'}(t+1) = \mathbf{r}_1(\mathbf{P}^*(t) + \mathbf{k})$
      Compare $\mathbf{P}(t+1)$ and $\mathbf{P'}(t+1)$ to select best solution $\mathbf{P}^*$
      if best fitness is not changed for last two iterations **then**
        Mutate solution as $\mathbf{P}(t+1) = \mathbf{k} * z^2 - h\frac{cos(x)}{1-cos(x)}$
      end if
    end for
    Update the fitness function F_n for each $\mathbf{P}(t)$ using BER
  else
    Update agents' position using sine-cosine optimization method.
  end if
  Update BER and SC parameters, $t = t + 1$
end while
Return $\mathbf{P}^*$

- max_depth: The highest number of levels per decision tree. A deeper tree can capture more complex patterns in the training data but also increase overfitting chances. The depth has to be controlled to have a well-balanced model [31].
- min_samples_split: The minimum number of data points that must be present on a node before it is split. Establishing an optimal concentration plot precludes the creation of more nodes with insufficient data points that contribute to generalization [32].
- min_samples_leaf: The leaf node's minimum number of data points. Determining a minimum leaf size limits the tree's coarseness and prevents it from being too fine-tuned to training data [33].
- Bootstrap: This parameter specifies whether sampling is with replacement or without. The selection between these methods affects the variety of trees in the forest [34].

Several combinations of these hyperparameter values will be searched out in the hyperparameter tuning process, and a performance test will be performed over a chosen evaluation metric [35], [36]. The outcomes of such a tuning procedure presented in the Experimental Results section will permit us to assess a fine-tuned configuration that improves performance characteristics for predictive capabilities [37].

DOI: https://doi.org/10.54216/JAIM.100104

Received: March 28, 2025 Revised: June 18, 2025 Accepted: August 19, 2025

4 Experimental Results

This section provides a comprehensive illustration of the empirical evidence from this research study in support of our collaborative framework, which integrates advanced machine learning models with sophisticated optimization mechanisms to enhance security. They concentrate on the BBERSC algorithm as a crucial aspect of improving how network requests are classified based on their accuracy. We carefully evaluate the quality of different classifiers such as Logistic Regression, Decision Tree, Random Forest Classifier, and Support Vector Machine classifier.

Table 1 shows in more depth the performance of basic classification models on a dataset used for testing. The main performance indicators of accuracy investigation are sensitivity, specificity, PPV, and NPV for all algorithms. Moreover, the Random Forest Classifier can be distinguished by a relatively high accuracy of 81%, which is better than other models. The table provides a general overview in which the strengths and weaknesses of each classification algorithm are presented in terms of network security.

Table 1: Results of the basic classification models on the tested dataset

Classification	Accuracy	Sensitivity	Specificity	PPV	NPV	F-Score
SGDClassifier	0.630	0.163	0.982	0.875	0.275	0.609
LogisticRegression	0.715	0.663	0.754	0.671	0.667	0.748
DecisionTreeClassifier	0.735	0.756	0.719	0.670	0.710	0.796
GaussianNB	0.745	0.640	0.825	0.733	0.683	0.752
KNeighborsClassifier	0.785	0.640	0.895	0.821	0.719	0.767
SVM	0.785	0.581	0.939	0.877	0.699	0.748
RandomForestClassifier	0.810	0.628	0.947	0.900	0.740	0.771

Figure 4 presents how various classification algorithms are performed. The figure represents the results of the accuracy investigation to evaluate their performance metrics in comparison with each other. The visual graph allows one to quickly interpret how each algorithm performs with regard to accuracy, sensitivity, specificity, and other significant measures. The graphical representation improves the usability of results, allowing researchers and practitioners to locate trends and patterns immediately. This diagram is a beneficial visual representation that facilitates a better understanding of the comparison between performance results from different classification models in network security.

DOI: https://doi.org/10.54216/JAIM.100104

Received: March 28, 2025 Revised: June 18, 2025 Accepted: August 19, 2025

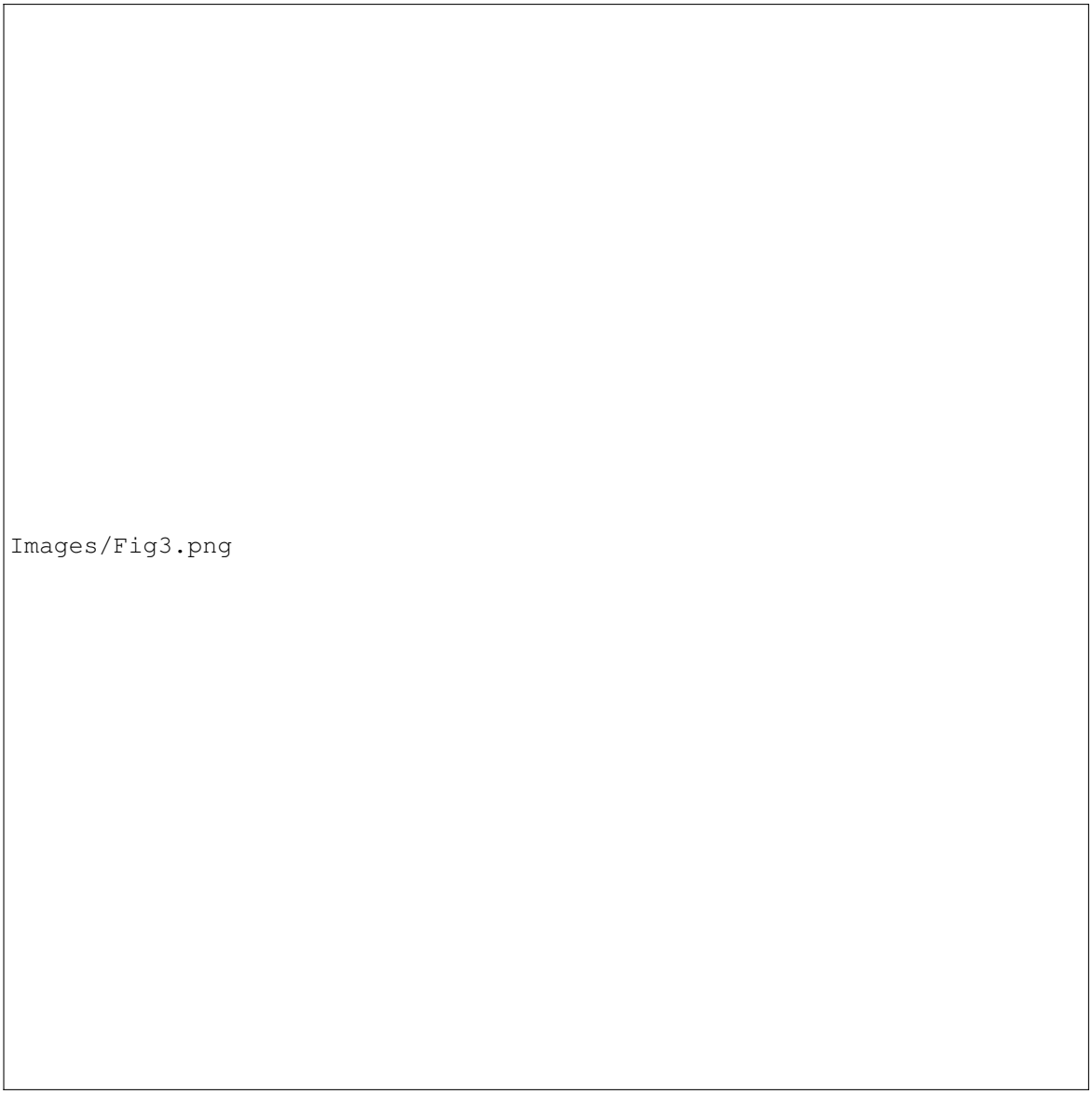

Figure 4: Accuracy investigation of various classification algorithms.

Table 2 presents a comprehensive performance comparison of various optimization algorithms integrated with the Random Forest Classifier for network requests classification. The results demonstrate the superior performance of the proposed BBERSC algorithm, which achieves the highest accuracy of 99.0%, significantly outperforming all other metaheuristic algorithms tested. BBERSC also exhibits exceptional performance across all evaluation metrics, with sensitivity (99.3%), specificity (98.8%), positive predictive value (98.8%), negative predictive value (99.3%), and F-Score (99.0%) all exceeding 98%. The individual component algorithms, BER and SCA, achieve accuracies of 96.6% and 95.4% respectively, indicating that their hybrid combination in BBERSC provides substantial performance enhancement. Among the baseline algorithms, PSO demonstrates moderate performance with 94.0% accuracy, while GWO (92.2%) and WOA (91.1%) show declining effectiveness. The traditional Genetic Algorithm (GA) exhibits the lowest performance with 88.2% accuracy, highlighting the advantage of modern metaheuristic approaches. The consistent superiority of BBERSC across all performance metrics validates the effectiveness of balancing exploration and exploitation capabilities through the hybrid optimization strategy, making it particularly suitable for complex network security classification tasks.

Figure 5 empirically studies the accuracy of various optimization algorithms to compare their performance when combined with a Random Forest Classifier. The presented illustration allows researchers and

DOI: https://doi.org/10.54216/JAIM.100104

Received: March 28, 2025 Revised: June 18, 2025 Accepted: August 19, 2025

Table 2: Performance evaluation of various optimization algorithms with the Random Forest Classifier.

Model	Accuracy	Sensitivity	Specificity	PPV	NPV	F-Score
BBERSC	0.990	0.993	0.988	0.988	0.993	0.990
BER	0.966	0.969	0.964	0.964	0.969	0.966
SCA	0.954	0.950	0.958	0.957	0.951	0.954
PSO	0.940	0.930	0.950	0.950	0.930	0.940
GWO	0.922	0.915	0.931	0.941	0.901	0.928
WOA	0.911	0.909	0.913	0.935	0.881	0.922
GA	0.882	0.879	0.887	0.924	0.825	0.901

practitioners to spot trends and patterns easily. This figure also contributes to the overall understanding of how various optimization algorithms impact general classification accuracy and informs about each algorithm's relative strengths and weaknesses in terms of improving network security.

Figure 5: Accuracy investigation of various optimization algorithms.

Figure 6 presents the accuracy histogram for differing optimization algorithms, which means a distribution value of their output. The accuracy distribution plot visualizes how consistent and variable each optimization algorithm is. From this figure, researchers can determine the distribution of accuracy values and central tendencies, which helps analyze the stability and reliability of the optimization models.

Figure 6: Investigation of the histogram of accuracy for various optimization algorithms.

Figure 7 shows the obtained residual values and heatmap of the BBERSC algorithm along with compared algorithms. The heatmap provides a visual representation of the residual values, which enables researchers to determine their strengths and weaknesses. Visualizing residual values improves the interpretability regarding algorithmic performance, allowing for a more detailed analysis of how closely the proposed BBERSC algorithm matches or differs from what is generally expected.

Figure 7: Residual Values and Heatmap for the proposed BBERSC algorithm and compared algorithms.

Figure 8 below shows the Receiver Operating Characteristic (ROC) curve for the BBERSC model. The ROC curve in a pictorial form indicates the trade-off between sensitivity and specificity that can give an overall picture of how well the algorithm performs at various classification thresholds. Researchers can use this figure to evaluate the discriminatory ability of the BBERSC algorithm in distinguishing safe and unsafe network requests.

Figure 8: ROC curve for the proposed BBERSC algorithm.

DOI: https://doi.org/10.54216/JAIM.100104

Received: March 28, 2025 Revised: June 18, 2025 Accepted: August 19, 2025

ANOVA statistical outcomes of the BBERSC algorithm and compared algorithms are shown in Table 3. A table that denotes the values of the Sum of Squares, Degrees of Freedom, mean square, F-statistic degrees and freedom, as well as P value for treatment effects and residuals, are presented below [38]. These statistical indices measure the relative importance of disparities among algorithms and provide researchers with evidence for whether or not a specific BBERSC algorithm is better than others, as well as to prove that this specification does indeed hold.

Table 3: ANOVA statistical results based on the BBERSC and compared algorithms.

	Sum of Squares	Degrees of Freedom	Mean Squares	F (DFn, DFd)	P value
Treatment	0.148	6	0.02467	F (6, 133) = 764.3	P<0.0001
Residual	0.004293	133	3.23E-05		
Total	0.1523	139			

Table 4 presents the output of the Wilcoxon Signed Rank Test used to analyze the BBERSC algorithm against other optimization algorithms to assess its overall statistical significance. P values and the sign of significance pertain to appraising the assumed impact. The following table provides some valuable insights into the stability and accuracy of this BBERSC algorithm that can be compared with other optimization models.

Table 4: Wilcoxon Signed Rank Test statistical results for BBERSC and other algorithms.

	BBERSC	BER	SC	PSO	GWO	WOA	GA
P value	.002	.002	.002	.002	.002	.002	.002
Exact/estimate?	Exact	Exact	Exact	Exact	Exact	Exact	Exact
alpha=.05?	Significant	Significant	Significant	Significant	Significant	Significant	Significant
Discrepancy	0.9902	0.9665	0.9539	0.9401	0.9222	0.9109	0.8824

Figure 9 shows a pair plot with regression lines for the proposed BBERSC algorithm to facilitate visual analysis of variable relationships. The pair plot deepens understanding of the algorithm's internal dynamics and relations between various parameters. This graph will help researchers understand the patterns and interactions that occur within the BBERSC algorithm based on which its performance can be interpreted. Each of the figures and tables offers an insightful yet concise perspective on a specific aspect or facet of the experimental results such that a set collective view is developed to present researchers with in-depth information about the strengths, weaknesses, aspects, inherent limitations, and implications implied by applying an integrated framework for Network Security.

Figure 9: Pair plot with Regression Lines for the proposed BBERSC algorithm.

5 Conclusion and Future Directions

At the end of this study, the BBERSC algorithm is integrated with various machine learning models as a new approach to strengthening network security. The presented general framework that includes Logistic Regression, Decision Tree, Random Forest, and Support Vector Machine; K-Nearest Neighbors, Naive Bayes, and SGD Classifier significantly improves accuracy. Of special note is the Random Forest model, which has a high accuracy of 0.99017198. Statistical validations, namely ANOVA and Wilcoxon Signed Rank Test, substantiate the recognition of the suggested solutions. Looking forward, the study indicates some promising avenues for future studies. Real-life implementation and scaling issues act as other crucial paths, guaranteeing the adaptability of the proposed framework in changing environments. Adding dynamic data sets representative of real-world cases will play an essential role in improving the model's adaptability to new cyber-threats. Future work should involve studying ensemble techniques, allowing various models to cooperate and develop resistant intrusion detection systems. Moreover, continued refinement and development

DOI: https://doi.org/10.54216/JAIM.100104

Received: March 28, 2025 Revised: June 18, 2025 Accepted: August 19, 2025

of optimization algorithms may improve the capacity of the proposed network security framework. This study provides the basis for future studies investigating adaptive and intelligent intrusion detection systems. However, cyber threats continue to change, and thus, network security innovations must be continually developed. The findings of this study are not only useful in terms of developing theories, but they also provide practical solutions that lay the foundation for a more proactive and sophisticated approach to network security against cyberattacks.

Availability of Data and Materials: The data that support the findings of this study are openly available on [kaggle.com] at [https://www.kaggle.com/code/nandinibagga/network-security-prediction-models/input]

References

[1] Y. Bengio, A. Lodi, and A. Prouvost, "Machine learning for combinatorial optimization: A methodological tour d'horizon," *European Journal of Operational Research*, vol. 290, no. 2, pp. 405–421, 2021, ISSN: 03772217. DOI: 10.1016/j.ejor.2020.07.063.

[2] D. S. Khafaga et al., "An Al-Biruni Earth Radius Optimization-Based Deep Convolutional Neural Network for Classifying Monkeypox Disease," en, *Diagnostics*, vol. 12, no. 11, p. 2892, Nov. 2022, ISSN: 2075-4418. DOI: 10.3390/diagnostics12112892. Accessed: Mar. 16, 2024. [Online]. Available: https://www.mdpi.com/2075-4418/12/11/2892.

[3] D. R. Nayak, R. Dash, B. Majhi, and S. Wang, "Combining extreme learning machine with modified sine cosine algorithm for detection of pathological brain," en, *Computers & Electrical Engineering*, vol. 68, pp. 366–380, May 2018, ISSN: 00457906. DOI: 10.1016/j.compeleceng.2018.04.009. Accessed: Mar. 16, 2024. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0045790617334572.

[4] V. Jackins, S. Vimal, M. Kaliappan, and M. Y. Lee, "AI-based smart prediction of clinical disease using random forest classifier and Naive Bayes," en, *The Journal of Supercomputing*, vol. 77, no. 5, pp. 5198–5219, May 2021, ISSN: 0920-8542, 1573-0484. DOI: 10.1007/s11227-020-03481-x. Accessed: Mar. 16, 2024. [Online]. Available: http://link.springer.com/10.1007/s11227-020-03481-x.

[5] C. M. Yeşilkanat, "Spatio-temporal estimation of the daily cases of COVID-19 in worldwide using random forest machine learning algorithm," en, *Chaos, Solitons & Fractals*, vol. 140, p. 110 210, Nov. 2020, ISSN: 09600779. DOI: 10.1016/j.chaos.2020.110210. Accessed: Mar. 16, 2024. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0960077920306068.

[6] T. Cuong-Le, T. Nghia-Nguyen, S. Khatir, P. Trong-Nguyen, S. Mirjalili, and K. D. Nguyen, "An efficient approach for damage identification based on improved machine learning using PSO-SVM," en, *Engineering with Computers*, vol. 38, no. 4, pp. 3069–3084, Aug. 2022, ISSN: 0177-0667, 1435-5663. DOI: 10.1007/s00366-021-01299-6. Accessed: Mar. 16, 2024. [Online]. Available: https://link.springer.com/10.1007/s00366-021-01299-6.

[7] R. Alkanhel et al., "Enhancing Wireless Sensor Network Efficiency through Al-Biruni Earth Radius Optimization," en, *Computers, Materials & Continua*, vol. 79, no. 3, pp. 3549–3568, 2024, Publisher: Tech Science Press, ISSN: 1546-2218, 1546-2226. DOI: 10.32604/cmc.2024.049582.

[8] Y. Tikhamarine, D. Souag-Gamane, A. Najah Ahmed, O. Kisi, and A. El-Shafie, "Improving artificial intelligence models accuracy for monthly streamflow forecasting using grey Wolf optimization (GWO) algorithm," en, *Journal of Hydrology*, vol. 582, p. 124 435, Mar. 2020, ISSN: 00221694. DOI: 10.1016/j.jhydrol.2019.124435. Accessed: Mar. 16, 2024. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0022169419311709.

[9] E. M. Hassib, A. I. El-Desouky, L. M. Labib, and E.-S. M. El-kenawy, "WOA + BRNN: An imbalanced big data classification framework using Whale optimization and deep neural network," en, *Soft Computing*, vol. 24, no. 8, pp. 5573–5592, Apr. 2020, ISSN: 1432-7643, 1433-7479. DOI: 10.1007/s00500-019-03901-y. Accessed: Mar. 16, 2024. [Online]. Available: http://link.springer.com/10.1007/s00500-019-03901-y.

DOI: https://doi.org/10.54216/JAIM.100104

Received: March 28, 2025 Revised: June 18, 2025 Accepted: August 19, 2025

[10] H. Nguyen, X.-N. Bui, Y. Choi, C. W. Lee, and D. J. Armaghani, "A Novel Combination of Whale Optimization Algorithm and Support Vector Machine with Different Kernel Functions for Prediction of Blasting-Induced Fly-Rock in Quarry Mines," en, *Natural Resources Research*, vol. 30, no. 1, pp. 191–207, Feb. 2021, ISSN: 1520-7439, 1573-8981. DOI: 10.1007/s11053-020-09710-7. Accessed: Mar. 16, 2024. [Online]. Available: https://link.springer.com/10.1007/s11053-020-09710-7.

[11] F. B. Banadkooki et al., "Enhancement of Groundwater-Level Prediction Using an Integrated Machine Learning Model Optimized by Whale Algorithm," en, *Natural Resources Research*, vol. 29, no. 5, pp. 3233–3252, Oct. 2020, ISSN: 1520-7439, 1573-8981. DOI: 10.1007/s11053-020-09634-2. Accessed: Mar. 16, 2024. [Online]. Available: http://link.springer.com/10.1007/s11053-020-09634-2.

[12] A. M. Anter, M. Abd Elaziz, and Z. Zhang, "Real-time epileptic seizure recognition using Bayesian genetic whale optimizer and adaptive machine learning," en, *Future Generation Computer Systems*, vol. 127, pp. 426–434, Feb. 2022, ISSN: 0167739X. DOI: 10.1016/j.future.2021.09.032. Accessed: Mar. 16, 2024. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0167739X21003782.

[13] S. Nusinovici et al., "Logistic regression was as good as machine learning for predicting major chronic diseases," en, *Journal of Clinical Epidemiology*, vol. 122, pp. 56–69, Jun. 2020, ISSN: 08954356. DOI: 10.1016/j.jclinepi.2020.03.002. Accessed: Mar. 16, 2024. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0895435619310194.

[14] H. Lu and X. Ma, "Hybrid decision tree-based machine learning models for short-term water quality prediction," en, *Chemosphere*, vol. 249, p. 126 169, Jun. 2020, ISSN: 00456535. DOI: 10.1016/j.chemosphere.2020.126169. Accessed: Mar. 16, 2024. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0045653520303623.

[15] Y. Ao, H. Li, L. Zhu, S. Ali, and Z. Yang, "The linear random forest algorithm and its advantages in machine learning assisted logging regression modeling," en, *Journal of Petroleum Science and Engineering*, vol. 174, pp. 776–789, Mar. 2019, ISSN: 09204105. DOI: 10.1016/j.petrol.2018.11.067. Accessed: Mar. 16, 2024. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0920410518310635.

[16] D. A. Pisner and D. M. Schnyer, "Support vector machine," en, in *Machine Learning*, Elsevier, 2020, pp. 101–121, ISBN: 9780128157398. DOI: 10.1016/B978-0-12-815739-8.00006-7. Accessed: Mar. 16, 2024. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/B9780128157398000067.

[17] M. Bansal, A. Goyal, and A. Choudhary, "A comparative analysis of K-Nearest Neighbor, Genetic, Support Vector Machine, Decision Tree, and Long Short Term Memory algorithms in machine learning," en, *Decision Analytics Journal*, vol. 3, p. 100 071, Jun. 2022, ISSN: 27726622. DOI: 10.1016/j.dajour.2022.100071. Accessed: Mar. 16, 2024. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S2772662222000261.

[18] A. Asadikia, A. Rajabifard, and M. Kalantari, "Systematic prioritisation of SDGs: Machine learning approach," en, *World Development*, vol. 140, p. 105 269, Apr. 2021, ISSN: 0305750X. DOI: 10.1016/j.worlddev.2020.105269. Accessed: Mar. 16, 2024. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0305750X2030396X.

[19] J. Zhou, A. H. Gandomi, F. Chen, and A. Holzinger, "Evaluating the Quality of Machine Learning Explanations: A Survey on Methods and Metrics," en, *Electronics*, vol. 10, no. 5, p. 593, Mar. 2021, ISSN: 2079-9292. DOI: 10.3390/electronics10050593. Accessed: Mar. 16, 2024. [Online]. Available: https://www.mdpi.com/2079-9292/10/5/593.

[20] H. Elazhary, "Internet of Things (IoT), mobile cloud, cloudlet, mobile IoT, IoT cloud, fog, mobile edge, and edge emerging computing paradigms: Disambiguation and research directions," en, *Journal of Network and Computer Applications*, vol. 128, pp. 105–140, Feb. 2019, ISSN: 10848045. DOI: 10.1016/j.jnca.2018.10.021. Accessed: Mar. 16, 2024. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S1084804518303497.

[21] A. Yassine, S. Singh, M. S. Hossain, and G. Muhammad, "IoT big data analytics for smart homes with fog and cloud computing," en, *Future Generation Computer Systems*, vol. 91, pp. 563–573, Feb. 2019, ISSN: 0167739X. DOI: 10.1016/j.future.2018.08.040. Accessed: Mar. 16, 2024. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0167739X18311099.

[22] M. Malawski, A. Gajek, A. Zima, B. Balis, and K. Figiela, "Serverless execution of scientific workflows: Experiments with HyperFlow, AWS Lambda and Google Cloud Functions," en, *Future Generation Computer Systems*, vol. 110, pp. 502–514, Sep. 2020, ISSN: 0167739X. DOI: 10.1016/j.future.2017.10.029. Accessed: Mar. 16, 2024. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0167739X1730047X.

[23] R. O. Aburukba, M. AliKarrar, T. Landolsi, and K. El-Fakih, "Scheduling Internet of Things requests to minimize latency in hybrid Fog–Cloud computing," en, *Future Generation Computer Systems*, vol. 111, pp. 539–551, Oct. 2020, ISSN: 0167739X. DOI: 10.1016/j.future.2019.09.039. Accessed: Mar. 16, 2024. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0167739X18303327.

[24] Y. Lu and X. Xu, "Cloud-based manufacturing equipment and big data analytics to enable on-demand manufacturing services," en, *Robotics and Computer-Integrated Manufacturing*, vol. 57, pp. 92–102, Jun. 2019, ISSN: 07365845. DOI: 10.1016/j.rcim.2018.11.006. Accessed: Mar. 16, 2024. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0736584518302801.

[25] L. Liu, C. Chen, Q. Pei, S. Maharjan, and Y. Zhang, "Vehicular Edge Computing and Networking: A Survey," en, *Mobile Networks and Applications*, vol. 26, no. 3, pp. 1145–1168, Jun. 2021, ISSN: 1383-469X, 1572-8153. DOI: 10.1007/s11036-020-01624-1. Accessed: Mar. 16, 2024. [Online]. Available: https://link.springer.com/10.1007/s11036-020-01624-1.

[26] L. J. Muhammad, E. A. Algehyne, S. S. Usman, A. Ahmad, C. Chakraborty, and I. A. Mohammed, "Supervised Machine Learning Models for Prediction of COVID-19 Infection using Epidemiology Dataset," en, *SN Computer Science*, vol. 2, no. 1, p. 11, Feb. 2021, ISSN: 2662-995X, 2661-8907. DOI: 10.1007/s42979-020-00394-7. Accessed: Mar. 16, 2024. [Online]. Available: http://link.springer.com/10.1007/s42979-020-00394-7.

[27] I. Sreeram and V. P. K. Vuppala, "HTTP flood attack detection in application layer using machine learning metrics and bio inspired bat algorithm," en, *Applied Computing and Informatics*, vol. 15, no. 1, pp. 59–66, Jan. 2019, ISSN: 22108327. DOI: 10.1016/j.aci.2017.10.003. Accessed: Mar. 16, 2024. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S2210832717301655.

[28] M. A. Hassan et al., "Evaluation of energy extraction of PV systems affected by environmental factors under real outdoor conditions," en, *Theoretical and Applied Climatology*, vol. 150, no. 1-2, pp. 715–729, Oct. 2022, ISSN: 0177-798X, 1434-4483. DOI: 10.1007/s00704-022-04166-6. Accessed: Mar. 16, 2024. [Online]. Available: https://link.springer.com/10.1007/s00704-022-04166-6.

[29] W. Tong, Q. Wei, H.-Y. Yan, M.-G. Zhang, and X.-M. Zhu, "Accelerating inverse crystal structure prediction by machine learning: A case study of carbon allotropes," en, *Frontiers of Physics*, vol. 15, no. 6, p. 63 501, Dec. 2020, ISSN: 2095-0462, 2095-0470. DOI: 10.1007/s11467-020-0970-8. Accessed: Mar. 16, 2024. [Online]. Available: https://link.springer.com/10.1007/s11467-020-0970-8.

[30] H. Li et al., "Machine learning assisted predicting and engineering specific surface area and total pore volume of biochar," en, *Bioresource Technology*, vol. 369, p. 128 417, Feb. 2023, ISSN: 09608524. DOI: 10.1016/j.biortech.2022.128417. Accessed: Mar. 16, 2024. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0960852422017503.

[31] G. Li, Y. Sun, and C. Qi, "Machine learning-based constitutive models for cement-grouted coal specimens under shearing," en, *International Journal of Mining Science and Technology*, vol. 31, no. 5, pp. 813–823, Sep. 2021, ISSN: 20952686. DOI: 10.1016/j.ijmst.2021.08.005. Accessed: Mar. 16, 2024. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S2095268621000902.

[32] K. Sandunil, Z. Bennour, H. Ben Mahmud, and A. Giwelli, "Effects of Tuning Hyperparameters in Random Forest Regression on Reservoir's Porosity Prediction. Case Study: Volve Oil Field, North Sea," in *All Days*, Atlanta, Georgia, USA: ARMA, Jun. 2023, ARMA–2023–0660. DOI: 10.56952/ARMA-2023-0660. Accessed: Mar. 16, 2024. [Online]. Available: https://onepetro.org/ARMAUSRMS/proceedings/ARMA23/All-ARMA23/ARMA-2023-0660/532467.

[33] M. H. D. M. Ribeiro, R. G. Da Silva, S. R. Moreno, V. C. Mariani, and L. D. S. Coelho, "Efficient bootstrap stacking ensemble learning model applied to wind power generation forecasting," en, *International Journal of Electrical Power & Energy Systems*, vol. 136, p. 107 712, Mar. 2022, ISSN: 01420615. DOI: 10.1016/j.ijepes.2021.107712. Accessed: Mar. 16, 2024. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0142061521009376.

[34] E. Elgeldawi, A. Sayed, A. R. Galal, and A. M. Zaki, "Hyperparameter Tuning for Machine Learning Algorithms Used for Arabic Sentiment Analysis," en, *Informatics*, vol. 8, no. 4, p. 79, Nov. 2021, ISSN: 2227-9709. DOI: 10.3390/informatics8040079. Accessed: Mar. 16, 2024. [Online]. Available: https://www.mdpi.com/2227-9709/8/4/79.

[35] P. Schratz, J. Muenchow, E. Iturritxa, J. Richter, and A. Brenning, "Hyperparameter tuning and performance assessment of statistical and machine-learning algorithms using spatial data," en, *Ecological Modelling*, vol. 406, pp. 109–120, Aug. 2019, ISSN: 03043800. DOI: 10.1016/j.ecolmodel.2019.06.002. Accessed: Mar. 16, 2024. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0304380019302145.

[36] H. A. Fayed and A. F. Atiya, "Speed up grid-search for parameter selection of support vector machines," en, *Applied Soft Computing*, vol. 80, pp. 202–210, Jul. 2019, ISSN: 15684946. DOI: 10.1016/j.asoc.2019.03.037. Accessed: Mar. 16, 2024. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S1568494619301632.

[37] R. Turner et al., "Bayesian Optimization is Superior to Random Search for Machine Learning Hyperparameter Tuning: Analysis of the Black-Box Optimization Challenge 2020," en, in *Proceedings of the NeurIPS 2020 Competition and Demonstration Track*, PMLR, Aug. 2021, pp. 3–26. Accessed: Mar. 16, 2024. [Online]. Available: https://proceedings.mlr.press/v133/turner21a.html.

[38] A. Djaafari et al., "Hourly predictions of direct normal irradiation using an innovative hybrid LSTM model for concentrating solar power projects in hyper-arid regions," en, *Energy Reports*, vol. 8, pp. 15 548–15 562, Nov. 2022, ISSN: 23524847. DOI: 10.1016/j.egyr.2022.10.402. Accessed: Mar. 16, 2024. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S2352484722023381.

DOI: https://doi.org/10.54216/JAIM.100104

Received: March 28, 2025 Revised: June 18, 2025 Accepted: August 19, 2025