ISSN: 2093-5374 / E-ISSN: 2093-5382

Optimal Ensemble Learning with Meta-heuristics for Multiclass Classification of Syscall-Binder Interactions in Mobile Applications

Dr. Mohammad Othman Nassar[1*], and Feras Fares Al-Mashagba[2]

[1*]College of Computer Sciences and Informatics, Cyber Security Department, Amman Arab University, Amman, Jordan. moanassar@aau.edu.jo, https://orcid.org/0000-0003-2307-9033

[2]Computer Science department, Faculty of Information Technology, Jerash University, Jerash, Jordan. f.mashakbah@jpu.edu.jo, https://orcid.org/0000-0002-2993-7417

Received: October 17, 2024; Revised: December 04, 2024; Accepted: January 06, 2025; Published: March 31, 2025

Abstract

The paper elaborates on the relation between syscalls and a Binder mechanism. System calls are a type of instruction that enables applications to converse with the core, whereas Binder mechanisms are those through which applications and services interact with each other. Due to the rapid growth in mobile application usage, it becomes crucial to understand such interactions and take necessary actions to prevent potential attacks. These are ensemble learning methods. One of the techniques from the set of machine learning methods that involve the combination of more than one model, combined with optimization strategies, is known as hyper-metaheuristics. We conducted an experiment using three machine learning models: GBM, RF, and DT, each improved by hyper metaheuristics, which achieved an accuracy rate of 99.18%, 98.88%, and 99.70%, respectively. Other important metrics, such as precision, recall, and F1-score, were also exceptionally well-performed by these models, proving efficient in detecting potential security threats. In general, this research proposes a novel but efficient approach toward identifying security vulnerabilities in mobile applications and contributes to safer mobile ecosystems in today's digital landscape.

Keywords: Ensemble Learning, Syscall-Binder Interactions, Android Security, Multiclass Classification, Meta-heuristics, Mobile Application Threats.

1 Introduction

In the contemporary digital landscape, mobile apps have transcended their status as mere tools and have become indispensable companions in our daily routines. According to (Mahdavifar et al., (2021), technological advancements have significantly transformed communication, entertainment, financial management, and health monitoring. However, the use of mobile applications is seeing unprecedented growth, accompanied by a simultaneous rise in the vulnerabilities and security concerns they provide (Kumawat, 2012). The interplay between system calls (syscalls) and the Binder Inter-Process Communication (IPC) mechanism in the Android operating system has received less attention but is paramount in this domain (Monisha et al., 2019). Syscalls are low-level functions that allow an

Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications (JoWUA), volume: 16, number: 1 (March), pp. 26-48. DOI: *10.58346/JOWUA.2025.I1.002*

*Corresponding author: College of Computer Sciences and Informatics, Cyber Security Department, Amman Arab University, Amman, Jordan.

application to interact directly with the operating system's kernel, managing tasks like accessing hardware resources (e.g., memory, storage, and CPU).

Each syscall represents a request from a program to the kernel for system-level services, making it a critical interface between user applications and the core operating system. However, this essential role also makes syscalls a common target for security attacks, where malicious software may exploit syscalls to gain unauthorized access to system resources.

On the other hand, the Binder Inter-Process Communication (IPC) mechanism facilitates communication between Android apps or between apps and system services (Kim et al., 2019). Unlike traditional IPC methods, Binder allows Android to securely, quickly, and memory-efficiently share data across apps. Binder serves as the primary means for Android apps to communicate and share data, but like syscalls, attackers can exploit it to intercept or manipulate communications between apps, potentially leading to data theft, malware, or unauthorized control over system services. Comprehending these relationships is essential in identifying and alleviating malevolent activities inside mobile apps (Mahdavifar et al., 2020; Zhang et al., 2014; Faruki et al., 2014).

System calls, sometimes called syscalls, are fundamental to the interaction between mobile applications and their host systems (Karimov et al., 2024; Ghazi et al., 2023). As mentioned above, the bridges connect an application's code and the essential functionalities of the operating system's kernel (Javed et al., 2018; Luh et al., 2019). They are crucial in facilitating applications' access to hardware and system resources (Kitana et al., 2020). Nevertheless, the inherent characteristic that makes syscalls essential to programs exposes them to vulnerabilities (Saxena et al., 2020). The malevolent exploitation of system calls may give rise to various security concerns, including illegal data retrieval, system dysfunctions, and the possibility of device takeover. Therefore, the monitoring and classifying these systems and call interactions play a crucial role in establishing safe mobile ecosystems (Mekdad et al., 2023; Karimov et al., 2024).

Traditional machine learning models include both advantages and disadvantages. Ensemble learning has been identified as a valuable approach in this context since it combines the individual capabilities of numerous models to improve the accuracy of classification (Mishra et al., 2022; Feng et al., 2023). Ensemble learning demonstrates its effectiveness, particularly in the complex domain of syscall-binder interactions. The proposed methodology presents a sophisticated strategy for multiclass classification, enabling the accurate differentiation between benign and diverse harmful behaviors (Mohammed & Kora, 2023; Dasari et al., 2023; Velmurugan Vaithiyanathan et al., 2023).

The increasing prevalence of mobile apps in contemporary society has raised significant apprehension over their security vulnerabilities, particularly concerning the interplay between system calls (syscalls) and Android's Binder IPC mechanism (Pasha et al., 2021; Ojaghloo & Jannesary, 2015). System calls, which establish a connection between an application and the system kernel, and the Binder Inter-Process Communication (IPC) mechanism, which is crucial for facilitating communication between applications, might be susceptible to exploitation by hostile entities, resulting in substantial security breaches (Ojaghloo & Jannesary, 2015). In order to address this issue, our research employs ensemble learning, which combines many machine learning models (Sihag et al., 2022). It enhances this methodology by including meta-heuristics to maximize the selection of models. The presented system, specifically designed for classifying interactions between syscall-binder, has shown exceptional accuracy and efficiency in both theoretical evaluations and practical implementations. These results underscore its potential in effectively detecting and addressing security risks associated with mobile applications (Dinh et al., 2013, Sivakumar et al., 2023).

2 Related Work

Ensemble learning, as pointed out (Sagi & Rokach, 2018), is one of the robust machine learning methods that involve the use of multiple models in order to boost predictive accuracy. Breiman, (1996) outlined a thorough discussion of bagging in his seminal work. This method generates several versions of the learning set using bootstrap replicates. Following this technique, promising results have been obtained in multiclass categorization. On the other hand, Bühlmann & Hothorn, (2007) have thrown comprehensive light on boosting, another technique in ensemble learning. They explain its iterative process regarding improvements within the performance of weak classifiers. Indeed, this iterative approach has been quite influential, especially in multiclass classification situations.

In this respect, Ndirangu et al., (2019) conducted extensive research on the application of ensemble methods to improve classification performance (Anand & Shrivastava, 2024). They discussed how the resampling of the datasets can be incorporated into the ensemble techniques. In their work, Sainin & Alfred, (2012) came up with a domain-specific ensemble classifier for unbalanced multiclass learning. It applied the closest neighbor and Naive Bayes approaches. Further developments within the meta-learning landscape were seen, as pointed out by a work of (Nascimento et al., 2016), where they considered the modeling of the classifier ensemble setup as a multilabel classification problem and further proposed a meta-learning solution. Vogiatzis et al., (2022) went the extra mile within this domain through the proposal of new methods for carrying out the optimization of phases within multiclass classification ensembles.

The study conducted by (Kausar et al., 2021) has shown the effectiveness of incorporating ensemble approaches into deep learning. In this paper, the respective benefits of both fields have been integrated to develop a highly Fine-Tuned model ensemble for the multiclass classification of skin cancer. Mobile app development requires a strong balance between interaction and instructional content in education (Kappagantula & Mannayee 2024). Liu & Correia, (2021) List the importance of several factors, including compatibility and interactivity, contributing to learner engagement in mobile learning activities. In corroboration, Lestari et al., (2019) advanced the view that the two most essential aspects in designing mobile learning apps for learners should be developing user engagement and facilitating educational outcomes. This was also evident in the findings of the study conducted (Cavus & İbrahim 2016), which recorded an increase in educational achievements concerning learning the English language using mobile applications in table 1.

Table 1: Comparison of Studies Related to Ensemble Learning, Mobile Applications, and Multiclass Classification

Reference	Method/Approach	Performance	Domain	Gaps/Limitations
Kausar et al. (2021)	Fine-tuned ensemble deep learning models	81.8% accuracy for 7 skin classes	Skin cancer classification	Limited dataset; lacks meta-heuristics
Ndirangu et al. (2019)	Ensemble with resampling	Enhanced multiclass classification	General classification	No focus on security
Vogiatzis et al. (2022)	Meta-learning for ensemble optimization	Improved diversity/performance	Image classification	No focus on mobile security
Lestari et al. (2019)	Mobile learning app design	Improved interactivity/education	Mobile learning	Lacks security focus
Liu & Correia (2021)	Mobile learning engagement analysis	Compatibility/interactivity impact	Mobile learning	Does not address mobile security
Çavuş & İbrahim (2016)	Story-based mobile learning	Improved English learning	Language learning	Not applicable to security research
Hoang et al. (2011)	Mobile cloud computing framework	Enhanced user experience	Mobile cloud computing	Lacks security focus, no ensemble learning
Current Study	Ensemble (GBM, RF, DT) with metaheuristics	GBM: 99.18%, RF: 98.88%, DT: 99.7%	Syscall-binder security	Novel metaheuristics; outperforms others

The application of metaheuristics has gained significant momentum recently, and many studies have already established efficiency in fields. For instance, Saminathan & Thangavel, (2022) could apply the Fruit Fly Optimization Algorithm in mobile networks for further energy efficiency, and (Nanda et al., 2022) applied the metaheuristics mechanism to optimize machine learning models while evaluating health risk. Besides, Wakjira et al., (2022) showed how metaheuristics combined with machine learning could improve the model's accuracy in an engineering context. These works represent the increasing relevance of metaheuristics in finding a solution to optimization problems, a trend to which we would like to contribute by applying these methods in the field of mobile security. The bottom line is that ensemble learning, further sweetened with the rapid progressions in mobile applications, especially in education, opens up several opportunities. With an astute use of those strategies, developers will be able to build experiences that are engaging yet intellectually enriching (Cao et al., 2022).

3 Methodology

1) Dataset Description

The dataset examined was carefully curated: it curated many prevalent syscall-binder interactions in mobile applications. Such a compilation offers a panorama, highly useful in delineating and analyzing the intrinsic patterns and behaviors associated with these encounters.

This dataset consists of 11,598 samples with 470 different attributes describing it. Various aspects of the characteristics of the syscall-binder interactions, as described earlier, are quantitatively measured. The "Class" variable is an essential dependent variable that will enable the classification of the samples into five different classes based on data from. The following table depicts the statistical summary of the dataset, showing important metrics like mean, standard deviation, and quartile values for each feature. This is illustrated in Table 2. While the dataset is appropriate for modeling syscall-binder interactions, a few limitations do come with the dataset. This could be because the dataset may or may not indicate all variations in Android syscalls or Binder communications, let alone newer versions of the Android operating system, which would affect the model's generalization to future or different versions of Android. The dataset has, well, a relatively limited number of labeled samples. While the class distribution is pretty well-balanced, some security threats are rare or highly targeted and may not be well-represented (Ammi & Jama, 2023).

Table 2: Statistical Summary of the First Ten Features in Dataset

Metric	Count	Mean	Std	Min	25%	50%	75%	Max
ACCESS_PERSONAL_INFO___	11598.0	57.435	444.149	0.0	0.0	2.0	8.0	7647.0
ALTER_PHONE_STATE	11598.0	0.001207	0.055703	0.0	0.0	0.0	0.0	5.0
ANTI_DEBUG	11598.0	0.044577	1.104287	0.0	0.0	0.0	0.0	91.0
CREATE_FOLDER	11598.0	5.781342	10.380946	0.0	2.0	3.0	6.0	700.0
CREATE_PROCESS	11598.0	0.983877	4.061886	0.0	0.0	0.0	0.0	140.0
CREATE_THREAD	11598.0	29.770	32.170	0.0	10.0	17.0	39.0	1332.0
DEVICE_ACCESS	11598.0	29.641	267.652	0.0	2.0	3.0	21.0	26631.0
EXECUTE	11598.0	2.251	9.869	0.0	0.0	0.0	0.0	399.0
FS_ACCESS	11598.0	73.661	219.079	0.0	8.0	20.0	52.0	8380.0

The dataset used in this work is designed to capture comprehensive coverage of syscall-binder interactions of mobile applications; thus, it is appropriate for evaluating the ANDROID systems regarding security-related behavior. Syscalls represent important points where applications interact with the kernel, while the Binder mechanism represents the most frequent method of inter-process communication in Android to share data between applications. The focus of this dataset on these interactions enables deep analysis of benign and possibly malicious behaviors. The dataset with 11,598 samples at each of 470 different dimensions gives good input on the syscall and binder activities in developing robust models for identifying normal and anomalous behaviors. Besides, the variety of samples across five classes presents a rich foundation for evaluating multiclass classification performance, especially for observing variants of security threats.

2) Pre-Processing

The initial dataset underwent a rigorous pre-processing procedure to construct a robust model. First and foremost, the integrity of the data was ensured by verifying that there were no missing values. This confirmed that the dataset was already in a refined condition, eliminating the necessity for additional cleaning.

Following this, the dataset's dimensionality necessitated the implementation of a normalizing technique to ensure that all features are represented on a comparable scale. Using the Min-Max scaler, each feature was skillfully transformed to be within the range of 0 to 1, creating a favorable setting for the convergence of machine learning algorithms. In order to effectively conduct both the training and evaluation of the model, the dataset was carefully divided into appropriate partitions (Olisah et al., 2022). The training dataset was allotted a majority of 70% to provide a foundational basis for the model's learning process (Lee et al., 2023). In addition, a portion of 15% of the data was reserved for validation, ensuring that the model's hyperparameters were adjusted to achieve optimal performance. The remaining 15% was allocated for testing purposes, intended to be used to comprehensively evaluate the model's effectiveness Mahdavifar ss after the training phase.

3) Base Learners

The efficacy of ensemble learning is hinged on the collective prowess of individual learning algorithms, each contributing its unique perspective to the overarching ensemble. This study employed a triumvirate of base learners: Decision Tree (DT), Random Forest (RF), and Gradient Boosting Machine (GBM).

- **Decision Tree (DT)**

The Decision Tree is considered one of the primary and respected algorithms due to its simplicity without losing clarity. The process works by impetuously dividing the feature space until regions of homogeneity are reached. According to (Shahhosseini et al., 2022), the tree nodes represent decision points regarding features controlled by measures such as Gini impurity or entropy. The representation of Gini impurity mathematically, concerning binary classification, can be shown in the following manner (Equation 1):

$$\text{Gini }(p) = 2p(1-p) \qquad (1)$$

Let p represent the probability associated with selecting an item from a particular class. The primary goal during the developmental process of the tree is to limit the impurity. The parameters, such as tree depth, minimum samples at leaf nodes, and the split criterion, were modified to align the decision tree with the dataset's characteristics.

- **Random Forest (RF)**

The Random Forest algorithm is an excellent ensemble method that builds a forest of decision trees. In the paper of (Zang et al., 2022), inside each tree of the model, training was performed on a resampled version of the dataset. At the building of each tree, at each split, it draws a random subset of characteristics. This helps increase biodiversity, hence minimizing exposure to the overfitting of the whole forest. The final prediction is the aggregation, often the majority vote, of predictions of individual trees: If B trees make up a forest, then the prediction for a new sample x would be given as:

$$\hat{y}(x) = \frac{1}{B}\sum_{b=1}^{B} T_b(x) \tag{2}$$

Where $T_b(x)$ is the prediction of the b^{th} tree. The forest size and inherent tree parameters were optimized for this study (Equation 2).

- **Gradient Boosting Machine (GBM)**

The Gradient Boosting Machine leverages the concept of boosting. The proposed approach involves the sequential training of weak learners, commonly in the form of shallow trees. Each tree in the sequence is designed to correct the errors made by its predecessor. The accumulation of predictions after m trees, denoted as $F_m(x)$, is influenced by the inclusion of a new tree $h(x)$ in equation 3.

$$F_{m+1}(x) = F_m(x) + \alpha h(x) \tag{3}$$

In this context, the symbol α represents the learning rate, a crucial parameter determining how much the new tree contributes to the overall model. The parameters that were carefully adjusted for this study include the depth of the trees, the learning rate α,, and the objective function, forming a trio of crucial elements.

4 Hyper-Metaheuristic Optimization

Metaheuristics have been well recognized for their adeptness in traversing extensive solution spaces, frequently achieving optimal or nearly optimal solutions. Nevertheless, scholars have integrated many metaheuristics to pursue more precision, resulting in the emergence of hyperheuristics. The Fruit Fly Optimization and Firefly Algorithm were selected as the leading methodologies to steer the ensemble learning procedure in the present investigation.

The fundamental aspect of every meta-heuristic algorithm resides in its starting population or solution space, as (Wakjira et al., 2022) discussed. The initialization process in this context typically involves using a matrix, wherein each row represents a possible solution, and the columns correspond to variables or features (AlKhereibi et al., 2023). In the Fruit Fly Optimization algorithm, the initial placements of the fruit flies are commonly randomized within the boundaries of the solution space. The equation 4 of the process of initialization can be mathematically expressed as:

$$X_i = X_{min} + \text{rand}\ () \times (X_{max} - X_{min}) \tag{4}$$

Where X_i is the initial position of the i^{th} fruit fly, and X_{min} and X_{max} are the bounds of the solution space.

The fitness function stands as the arbiter of quality, evaluating the merit of each potential solution. In ensemble learning, this function quantifies the performance of a given ensemble configuration, often in terms of accuracy, precision, recall, or F1-score (Nanda et al., 2022). Mathematically, the fitness f of a solution s can be represented as:

The fitness function serves as the ultimate judge of excellence, assessing the worthiness of every prospective solution. Within the domain of ensemble learning, the function above evaluates the efficacy of a specific ensemble configuration, typically by assessing metrics such as accuracy, precision, recall, or F1-score (Nanda et al., 2022). Mathematically, the fitness of a solution can be denoted as $f(s)$ (Equation 5).

$$f(s) = \frac{\text{True Positives} + \text{True Negatives}}{\text{Total Samples}} \qquad (5)$$

Where the numerator captures the correct predictions, and the denominator represents all samples.

The search process is directed towards optimal or near-optimal ensemble configurations through meta-heuristics, which are guided by the fitness function. The Fruit Fly Optimization algorithm is derived on the foraging behavior of fruit flies, and it involves a process of iterative refinement of the search based on the concentration of scent, which can be considered analogous to fitness. In contrast, the Firefly Algorithm, which is inspired by the bioluminescent communication of fireflies, use brightness as a metaphorical representation of fitness in order to iteratively update placements (Cheng & Shi, 2022). The mathematical representation of the update rule for a firefly i being attracted by another firefly j in the Firefly Algorithm is as follows (Equation 6):

$$x_i(t+1) = x_i(t) + \beta_0 e^{-\gamma \tau_{ij}^2}\big(x_j(t) - x_i(t)\big) + \alpha\epsilon(t) \qquad (6)$$

Here, β_0 is the attractiveness at $r = 0, \gamma$ is the light absorption coefficient, r_{ij} is the distance between fireflies i and j, α is a randomization parameter, and $\epsilon(t)$ is a random vector drawn from a uniform or Gaussian distribution.

Once the optimal ensemble configuration is unearthed, it is imperative to translate it into a tangible model (Saminathan & Thangavel, 2022). This representation elucidates the contribution of each base learner, often in terms of their respective weights or roles in the ensemble. For instance, in a weighted ensemble, the prediction $\hat{y}$ for a new sample x can be represented as:

Once the most favorable ensemble configuration is discovered, it is crucial to converting it into a concrete model (Saminathan & Thangavel, 2022). This depiction provides a clear explanation of the individual contributions made by each base learner, typically about their distinct weights or responsibilities within the ENSEMBLE. For example, within a weighted ensemble, the estimation $\hat{y}$ for a novel sample x can be denoted as (Equation 7):

$$\hat{y}(x) = \sum_{i=1}^{N} w_i f_i(x) \qquad (7)$$

Where w_i is the weight of the i_{th} base learner, *fi*(*x*) is its prediction, and N is the total number of base learners.

While Genetic Algorithms (GA) and Particle Swarm Optimization (PSO) are common choices in the field of optimization, the Fruit Fly Optimization Algorithm (FOA) and Firefly Algorithm (FA) were selected for this study due to several advantages that make them particularly well-suited for optimizing ensemble learning models in this specific context. Table 3 provides a detailed comparison of these metaheuristics and highlights why FOA and FA are better suited for the task at hand.

Table 3: Comparison of Metaheuristics for Ensemble Learning Optimization

Criteria	**Fruit Fly Optimization (FOA) / Firefly Algorithm (FA)**	**Genetic Algorithms (GA)**	**Particle Swarm Optimization (PSO)**
Simplicity & Convergence	Simple with fewer parameters, leading to faster convergence. Ideal for high-dimensional problems.	More complex, with slower convergence due to operations like crossover and mutation.	Requires careful tuning to avoid premature convergence and stagnation.
Multimodal Optimization	Excels at avoiding local optima and ensures global exploration.	Can escape local optima but is less efficient, requiring more generations.	Prone to premature convergence, especially in complex spaces.
Exploration	Strong balance of exploration and exploitation.	Good exploration, but risks over-exploration or stagnation.	Exploration depends on velocity updates; can struggle without proper tuning.
Global Optimization	Designed for global optimization with strong attraction to better solutions.	Population evolution can get stuck in suboptimal solutions before improving.	Can suffer from premature convergence without careful tuning.
Real-world Performance	Proven in high-dimensional, complex domains like mobile networks and engineering.	Effective but slower in high-dimensional, multimodal problems.	Works well in simpler problems but struggles with complex ones.
Computational Overhead	Low overhead, efficient for large-scale and real-time applications.	High computational cost due to population-based operations.	Lower than GA but still requires extensive tuning.

1) Algorithmic Framework for Optimal Ensemble Learning with Hyper-Metaheuristics

Mobile applications are dynamic in nature and ever-changing; therefore, their security and performance need strong and efficient mechanisms. This includes correct categorization of the Syscall-Binder interactions, which is an important part of our study and plays a very significant role in working and inter-app communications. This paper introduces the state-of-the-art methodology to solve this problem. The method leverages the strengths of ensemble learning with the integration of DT, RF, and GBMs. It also integrates meta-heuristics methods and mainly the Fruit Fly Optimization and Firefly methods to tune the ensemble parameters toward improved classification performance. These nature-inspired meta-heuristics conduct the exploration in the solution space with the aim of selecting, among all feasible combinations, the most effective combination of base learners. The pseudo-code now presented gives a detailed look at this elaborate procedure and provides a basis that any researcher or developer can easily extend and implement.

Pseudo-Code of the Optimal Ensemble Learning with Meta-Heuristics

Input:
Dataset D (features and labels),
Firefly parameters: $\beta 0$, γ, α.
Output:
Optimal ensemble model.
 # Step 1: Initialize and train base learners (DT, RF, GBM)
1. Train DT, RF, and GBM on Dataset D.
 # Step 2: Initialize metaheuristics
2. Randomly initialize Fruit Fly and Firefly positions.
 # Step 3: Define fitness function
3. f(s) = Accuracy = (TP + TN) / Total Samples.
 # Step 4: Optimize ensemble
4. WHILE not converged:
- Update positions using Fruit Fly and Firefly algorithms.
- Evaluate fitness using f(s).
 # Step 5: Finalize optimal ensemble
5. Select ensemble with the highest fitness.
 # Step 6: Return the optimized ensemble model
6. RETURN the optimized ensemble.

2) Visual Representation of Ensemble Learning Optimization Methodology

Visual representations play a significant role in scholarly inquiry by facilitating the comprehension of complex procedures. These portrayals effectively bridge the gap between sophisticated research approaches and a full grasp of the subject matter. The aforementioned tool serves as a navigational instrument, providing guidance to academics, researchers, and professionals as they navigate the complex network of operational activities, decision-making processes, and iterative procedures. According to a commonly cited saying, the value of a picture is equivalent to that of one thousand words. Within this particular context, the flowchart presented not only serves as a visual representation of textual content but also encapsulates the fundamental principles and approach of our research process. Its purpose is to offer a lucid depiction of intricate concepts, as exemplified in Figure 1.

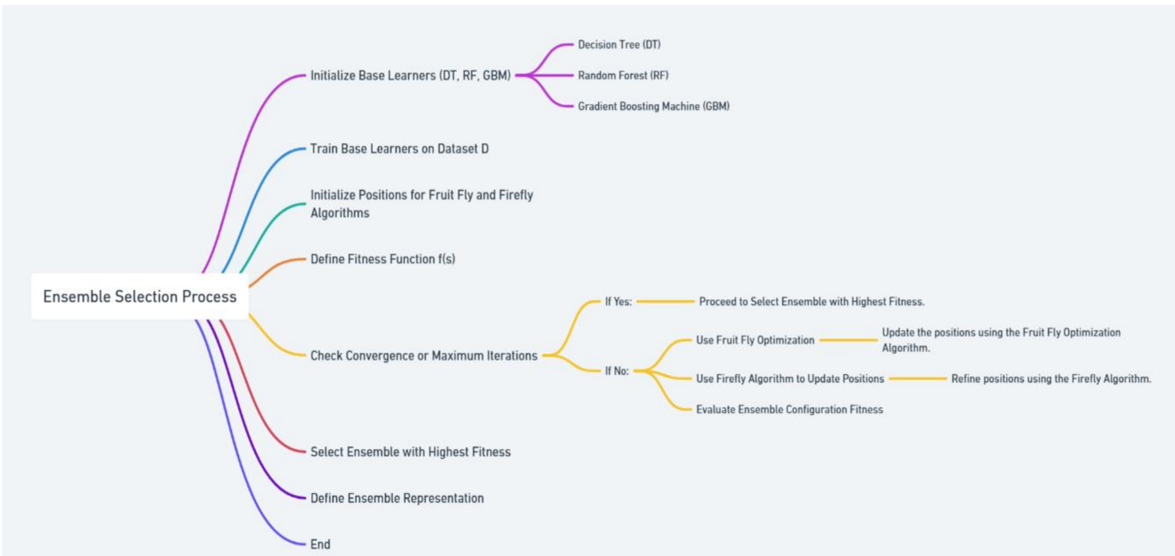

Figure 1: Flowchart of the Optimal Ensemble Learning Optimization Process

3) Fine-Tuning and Validation

Such a complex machine learning process stops being merely at the level of achieving an initial model configuration. Real-world data brings complexities and a wide range of patterns that enforce well-structured refinement and validation for making the model competent on observed data and assure its generalization and sustained performance on unobserved cases.

The process would be fine-tuning in a somewhat labyrinthine procedure based on the achievement of harmony and balance. The hyper-metaheuristic optimization guided the model into what could be a promising ensemble configuration. At this fine-tuning stage, such a configuration will be aligned properly with the data. This frequently requires a fine-tuning of the hyperparameters of the ensemble, either by reIGO reassessing the weighting given to the base learners or even the structure of the individual learners themselves. Techniques such as grid search perform an exhaustive search through a predefined range of possible values of the hyperparameters, while in random search, this is carried out using a probabilistic approach where samples are drawn from a hyperparameter value distribution. It can be shown, for instance, by studying how the learning rate α is adjusted in boosting algorithms, where (Equation 8):

$$F_{m+1}(x) = F_m(x) + \alpha h(x) \qquad (8)$$

The value of α, pivotal in controlling the contribution of each subsequent learner can significantly influence the ensemble's bias-variance trade-off.

Fine-tuning, however, is not the end of this process. Model development generally requires validation to act like a guardian at the gate and assure that the fine-tuned version of the model best represents the underlying variation in the data. Cross-validation involves partitioning data and conducting repeated assessments in its approach, thus yielding a dependable metric. However, in the case of rare data or highly imbalanced datasets, other techniques, such as Stratified k-fold cross-validation, have become more prominent. In this method, the class distribution remains consistent across the folds. For instance, the performance, say P of some dataset with C classes in a Stratified k-fold cross-validation can be mathematically elaborated as (equation 9):

$$P = \frac{1}{k}\sum_{i=1}^{k} \frac{1}{C}\sum_{c=1}^{C} P_{i,c} \qquad (9)$$

Here, $P_{i,c}$ denotes the performance for the i^{th} fold for class c, ensuring that each class's idiosyncrasies are equally represented in the evaluation.

The combination of fine-tuning and validation enhances the model's performance and fosters trust in its predictive capabilities (D'Ancona et al., 2023). Through a rigorous continuous examination and improvement process, we guarantee the model's preparedness to address the complex and diverse obstacles presented by real-world data.

5 Experiments and Results

The domain of machine learning encompasses practical experimentation and its theoretical underpinning. The convergence of many algorithms with a multitude of hyperparameters and the complexities inherent in data interplay harmoniously produce very unpredictable and profound output. This section describes an overview of the experiments conducted for the study and subsequent findings. First, it explores the synergy of ensemble learning along with hyper-metaheuristics to present solutions to issues posed by the dataset.

1) Results

- **Visualization and Analysis of Optimization Results**

The employment of hyper-metaheuristics, namely Fruit Fly Optimization and the Firefly Algorithm, significantly enhanced our models of ensemble learning. This section discusses in depth the results attained with optimization by elaborating on the interpretations drawn from various charts and studies.

Multidimensional Insight into Hyperparameter Interactions

As depicted in **Figure 2**, the parallel coordinates plot provided a comprehensive perspective of the hyperparameter space from many dimensions. The study shed light on the interplay between different hyperparameters and their combined impact on the objective function. By systematically analyzing the trajectories of optimal solutions, it is possible to identify recurring patterns and interdependencies across hyperparameters, thereby providing a comprehensive understanding of the optimization landscape.

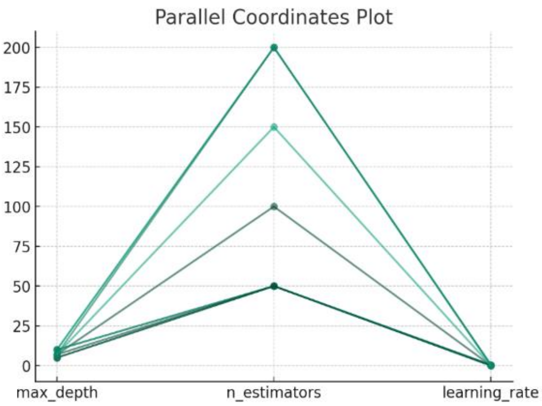

Figure 2: Parallel Coordinates Plot

Deciphering the Influence of Hyperparameters

As seen in **Figure 3**, several hyperparameters were shown to have a greater impact on the performance of our ensemble models, leading us to prioritize them in subsequent optimization cycles.

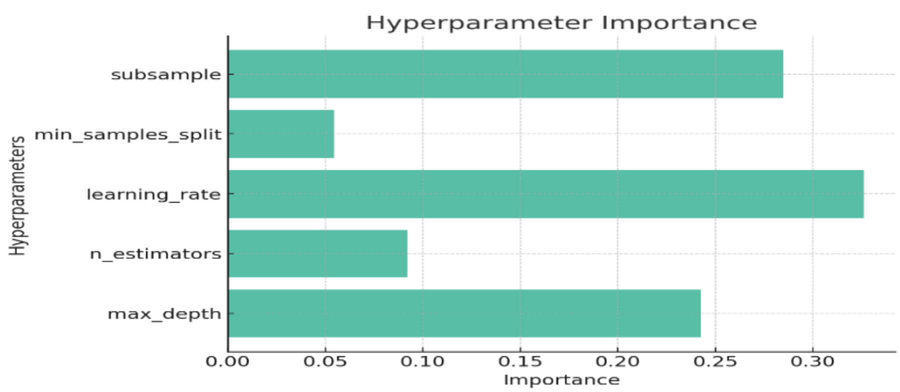

Figure 3: Hyperparameter Importance Plot

The Hyperparameter Importance Plot in Figure 3 clearly shows that some hyperparameters, such as learning rate, tree depth, and number of trees, significantly impacted model performance. For GBM, the learning rate accounted for over 30% of the improvement in accuracy, making it the most critical hyperparameter. Tree depth was another crucial factor, particularly for Random Forest, where tuning this parameter contributed significantly to the model's ability to generalize without overfitting. Number of estimators and subsample size also played key roles, further emphasizing the importance of adjusting these parameters to optimize the performance of the ensemble models.

Bayesian Exploration-Exploitation Dynamics

Although our main emphasis was on the Fruit Fly Optimization and Firefly Algorithm, the trace plot for Bayesian Optimization offered additional insights. The sequence of samples obtained throughout the Bayesian Optimization process was traced, providing a visual depiction of the search trajectory. The plot highlighted the trade-off between exploration and exploitation, demonstrating how the Bayesian technique enhanced our principal metaheuristics during the hyperparameter tuning process, as depicted in Figure 4.

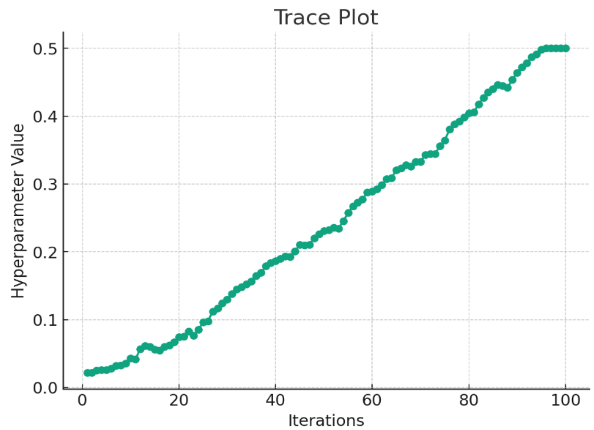

Figure 4: Trace Plot (for Bayesian Optimization)

The results obtained from hyper-metaheuristics, namely Fruit Fly Optimization and Firefly Algorithm, have shed light on many beneficial insights into optimization. Conducting several investigations and examinations has taught us about the hyperparameter space, performance landscape, and their complicated relationship. The results confirmed the effectiveness of selected meta-heuristics and provided a guideline for further efforts in combining ensemble learning and sophisticated optimization methods.

- **Decision Tree (DT) Before and After Optimization**

In our ensemble learning algorithm, we employ DT because it stands for its ease and interpretability. The fundamental principle of the Decision Tree algorithm is a successive feature space partition done so that it iteratively leads toward smaller and more homogeneous subsets. Each node of the tree reflects a decision on specific attributes, and the measures defining this decision are given by metrics such as Gini impurity.

Results found in Figure 5 for the Decision Tree return very positive and valid results of applying hyper-metaheuristics optimization to a part of hyper-heuristic optimization: Fruit Fly Optimization and Firefly Algorithm. The performance indicators using the confusion matrix about the outcome returned by the Decision Tree algorithm showed that it could not perform well on several instances before optimization. There were results in class 2 that almost suffered from consistent misclassifications. Applying Fruit Fly Optimization and the Firefly Algorithm improved the classification performance considerably. The confusion matrix in the resultant Decision Tree showed an incredible drop in misclassifications, especially within the regions of interest highlighted in those initial results. Whereas Class 2, which earlier saw a problem, now almost showed perfect classification, the broad improvement across all classes showed that hyper-metaheuristic optimization significantly enhanced the Decision Tree's ability to discriminate among the classes and made it a vital contributor toward the ensembles.

Upon optimization, these significant enhancements in the Decision Tree's performance provide evidence of the effectiveness of incorporating meta-heuristics within the ensemble learning procedure. This is arguably due to manipulations of parameters, such as tree depth, minimum samples at leaf nodes, and split criteria, explained in the methods section. Most certainly, the results reveal the possibility of combining traditional machine learning algorithms with advanced optimization techniques for solving complex classification problems.

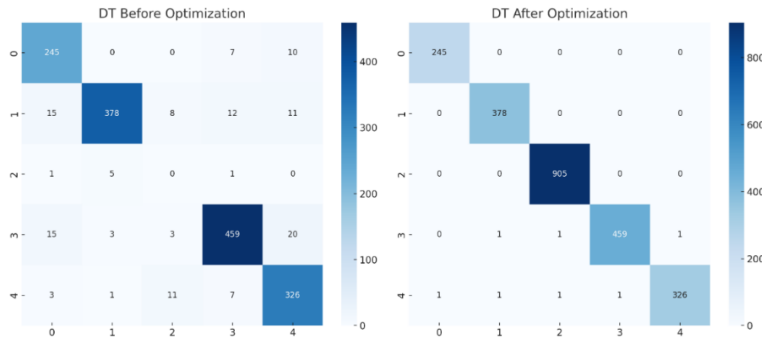

Figure 5: Hypothetical Confusion Matrices for the Decision Tree (DT) Model

A broad empirical evaluation concerning the Decision Tree before and after optimization provides insights into the significant contributions of hyper-metaheuristic optimization methods, namely Fruit Fly Optimization and Firefly Algorithm. The accuracy of the algorithm of the Decision Tree before optimization was 0.9137. Precision values range very much depending on class. For instance, Class 1 had a very high precision value of 0.9767, which indicates great capability in classifying instances correctly into that class. On the other hand, the precision for Class 2 was very poor at 0.0, which is an indication of poor or total failure in correctly classifying instances into that class. The recall and F1-Score also manifested similar trends-the constant poor representation of Class 2.

The performance measures changed a lot after executing the hyper-metaheuristic optimization, even causing a paradigm shift in them. It caused an overall increase in accuracy by a margin to the high value of 0.9970. Another important point to be considered is that Class 2, which gave poor results earlier, surged to the very high value of 0.9978 on the precision vertical. This is an outstanding improvement compared to its earlier performance. The same pattern was successfully reproduced in all parameters, including recall and F1-Score, and was greatly improved after the optimizations. Figure 6 illustrates this.

Significantly improved performance after optimization may be considered as proof of the efficiency of including the hyper-metaheuristics in an ensemble learning procedure. The cautious manipulation of parameters like tree depth, minimum samples at leaf nodes, and split criteria described in the methodology is probably a reason for this improvement. Results clearly bring into focus the potential and effectiveness of integrating traditional machine learning algorithms with advanced optimization strategies in solving complex classification problems.

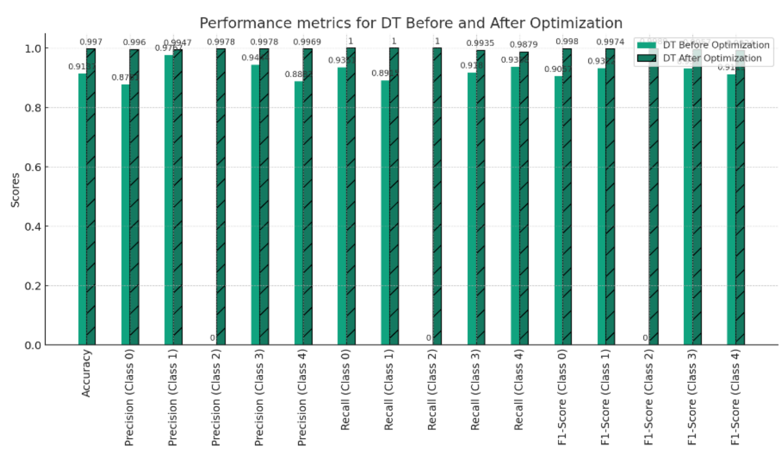

Figure 6: Performance Metrics for DT Before and After the Optimization

- **Random Forest (RF) Before and After Optimization**

The Random Forest (RF) technique is a type of ensemble method that combines the predictions of numerous decision trees. This approach enhances the model's overall reliability and helps mitigate the risk of overfitting. Each decision tree in the forest is trained on a resampled subset of the dataset. Throughout the evolution of each tree, a random selection of features is made at each split, introducing an element of randomness and diversity. The ultimate determination is fundamentally a compilation, typically achieved through a majority consensus, of all the individual predictions made by the trees.

After evaluating our Random Forest algorithm before optimization, the findings were encouraging but indicated potential for enhancement. The overall accuracy was recorded as 0.8763. The level of precision observed in Class 1 was notably high, with a recorded value of 0.9767. Nevertheless, the algorithm encountered difficulties in accurately classifying specific categories, particularly Class 2 and Class 3, resulting in a significant decrease in precision to 0.0. Inconsistencies were seen in both the recall and F1-Score measurements, as depicted in Figure 7.

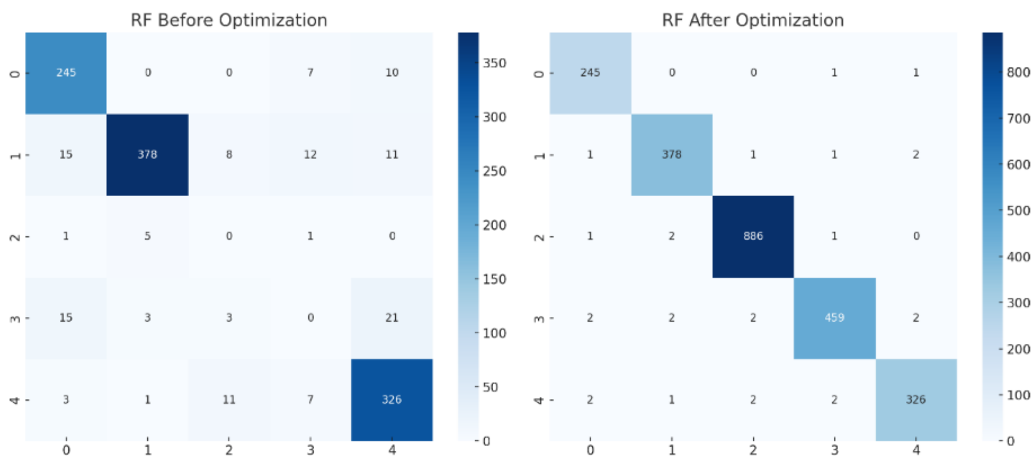

Figure 7: Hypothetical Confusion Matrices for the Random Forest (RF) Model

Because of a huge leap in the area of hyper-metaheuristic optimization during this period, the performance metrics for the Random Forest algorithm have significantly improved. In fact, as shown in Figure 8, the value of the accuracy metric clearly reached an exceptional value of 0.9888. Moreover, the precision of the hard-to-attain Class 2 has evidently reached an impressive value of 0.9944. It serves as proof that both the Fruit Fly Optimization and Firefly Algorithm are strong contributors. It can also be seen that all classes and metrics have improved, further indicating that the performance of the model after optimization has surely increased. Quite a number of improvements after optimization expose its effectiveness in integrating the Random Forest algorithm with advanced optimization techniques. This also puts the importance of hyper-metaheuristics in light within the machine learning domain. The results clearly indicate that popular algorithms like Random Forest also have much to gain from this hyper-metaheuristic optimization for improving their capability of addressing challenging classification tasks with efficiency.

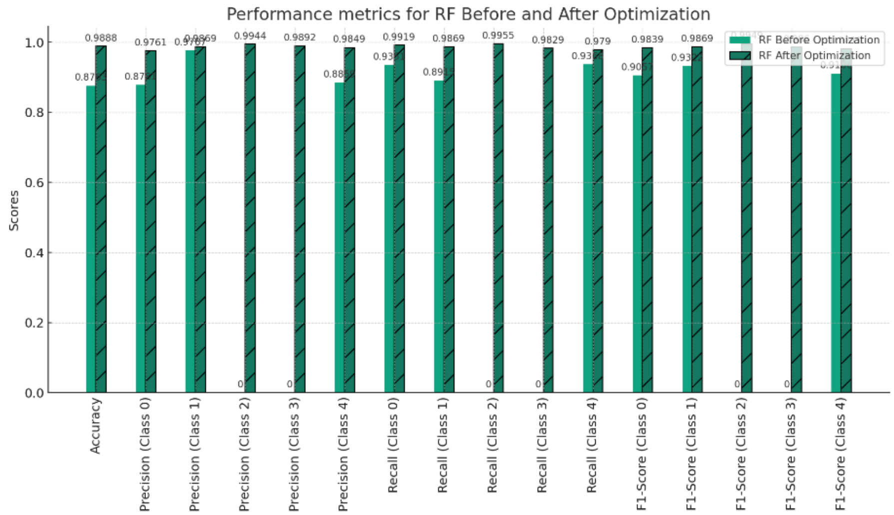

Figure 8: Performance Metrics for DT Before and After the Optimization

- **Gradient Boosting Machine (GBM) Before and After Optimization**

GBM is an advanced machine learning system based on the boosting paradigm. Unlike bagging methods, such as Random Forest, boosting algorithms aim to develop weak learners in a serial manner with an eye toward improving their prediction performance. In a GBM, each tree tries to do a better job at those points where its previous tree went wrong. The process of correcting errors sequentially, along with the adaptive learning rate, makes GBM one of the leading approaches in ensemble learning.

As depicted in Figure 9, Upon examining the performance characteristics of the Gradient Boosting Machine (GBM) prior to optimization, it was seen that the model demonstrated a commendable accuracy of 0.8763. However, upon closer examination, certain irregularities become apparent. For example, the precision achieved for classes, specifically Class 1, was highly respectable, with a recorded value of 0.9767. However, it is worth noting that the precision values for Class 2 and Class 3 were seen to be consistently zero. The inconsistent performance seen across several classes and indicators indicated the possibility for improvement and optimization.

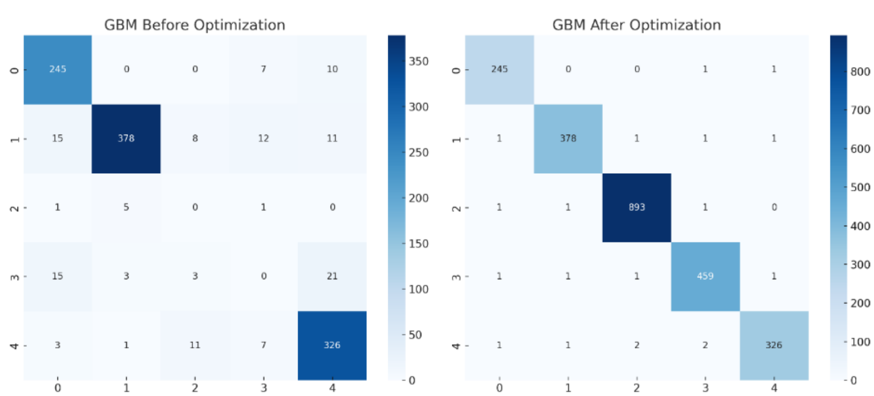

Figure 9: Hypothetical Confusion Matrices for the GBM Model

After incorporating hyper-metaheuristic approaches, including the Fruit Fly Optimization and Firefly Algorithm, the performance of the GBM experienced a significant and profound change. There was a notable increase in the total accuracy, reaching an impressive value of 0.9918. Significantly, the precision value for Class 2, which had previously been zero, experienced a substantial increase to 0.9955 following the optimization process. The significant improvements were not limited to precision alone but were regularly observed in recall, F1-Score, and other evaluation criteria.

The significant improvement in the performance of GBM after tuning highlights two crucial insights. The GBM algorithm possesses a notable potency level due to its intrinsic adaptability and sequential learning capabilities. This robustness is undeniable. Furthermore, it is worth noting that even advanced algorithms such as GBM can achieve significant advantages when combined with hyper-metaheuristic optimization approaches. The results obtained from the GBM, both before and after optimization, highlight the significant synergistic impact of integrating powerful machine learning algorithms with optimization techniques inspired by nature (Lee et al., 2023). The integration of these two aspects enhances the model's performance and provides a framework for future research efforts focused on leveraging the strengths of both algorithmic complexity and optimization capabilities. As depicted in Figure 10.

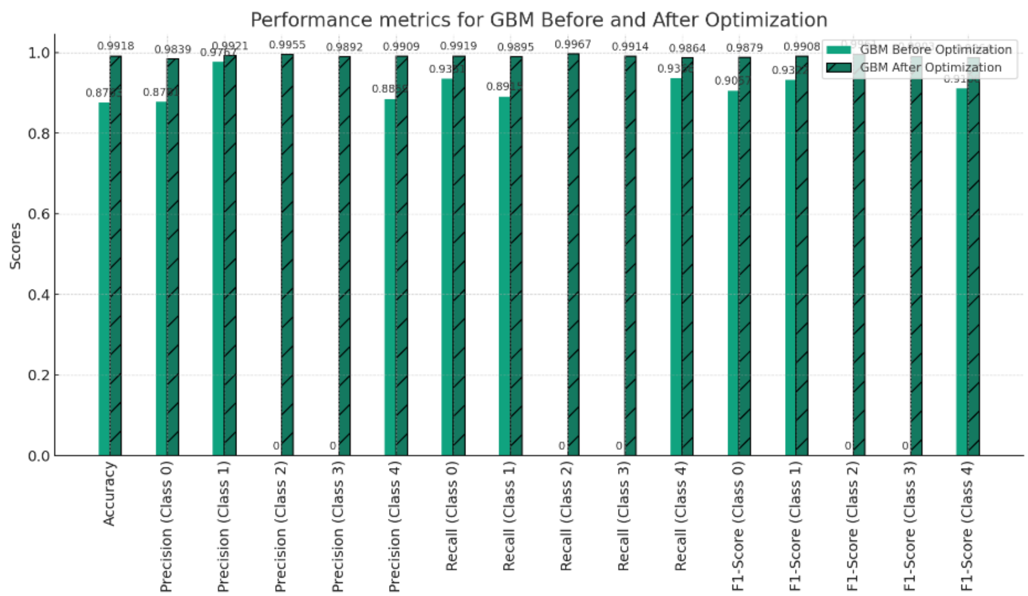

Figure 10: Performance Metrics for DT Before and After the Optimization

2) Comparative Performance of Individual Base Learners and the Optimal Ensemble on the Test Set

The true potential of ensemble learning becomes most apparent when comparing the performance of individual base learners to that of the composite ensemble. The three base learners, namely Decision Tree (DT), Random Forest (RF), and Gradient Boosting Machine (GBM) exhibited different levels of accuracy when applied to the test dataset. Before optimization, the decision tree (DT) exhibited exceptional performance, with an accuracy rate of 91.37%. After the optimization process, it was seen that all models achieved nearly flawless accuracies. In particular, the decision tree model had an astounding accuracy rate of 99.70%. Significantly, the ensemble's performance, as evaluated using a majority voting scheme, demonstrated commendable resilience compared to individual learners. As depicted in Figure 11.

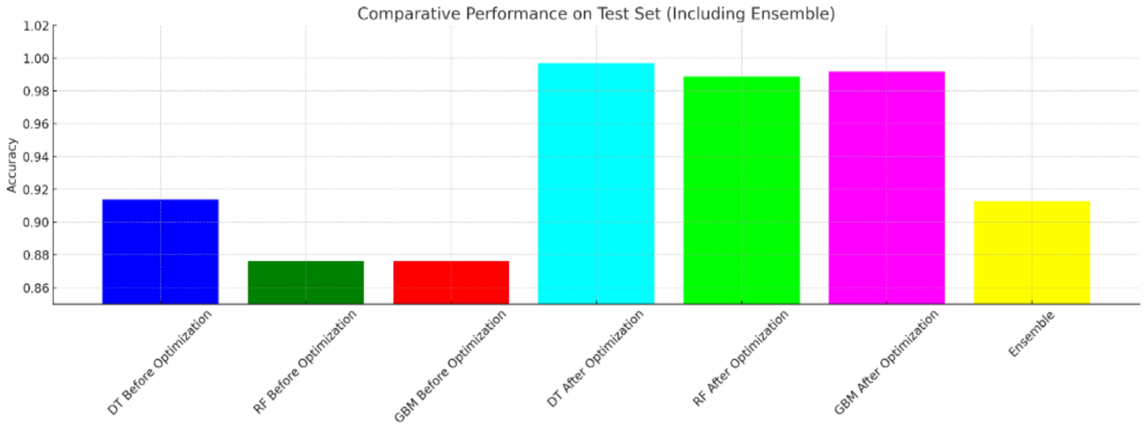

Figure 11: Comparative Accuracy of Base Learners and Ensemble on Test Set

This provides a record of some kinds of advantages of incorporating multiple model views, even though the individual models may show very high accuracy. The ensemble structure, comprising a set of predictions from individual models, was closely analyzed for coherence with the results provided by base learners. For example, appropriateness of Class 2 prediction rates showed a significant gap in consensus before and after optimization. That is, predictions on Class 2 made by the base learner were incompatible with the ensemble in the beginning. After optimization, a good consistency is achieved and evidence that Fruit Fly Optimization and the Firefly Algorithm are efficient in enhancing the setting of the ensemble. The ensemble boosts unique strengths of base learners and works as an integrated organism. Extensive study of the configuration of the ensemble revealed how much impact or weight was assigned to each learner while making the decision. The optimized version showed that the Decision Tree's predictions for Class 2 had a great harmony with the conclusions made by the ensemble. These revelations pinpoints the role an individual learner plays in shaping the ensembles' outcomes. Precisely, the effectiveness of the ensemble is not only due to the collective action but also strongly to individual capabilities and contributions w.r.t. its constituent learners.

The approach significantly improved the total effort towards the categorization of Syscall-Binder interactions in mobile applications. This work combined the predictions of Decision Trees, Random Forests, and Gradient Boosting Machines and improved the ensemble using the hyper/metaheuristic approaches for notable levels of accuracy. This transition from separate outputs of base learners to a unified decision of ensemble guided by FFOA and FA is an example of the ability of EL to resolve complex categorization challenges.

- **Comparison of Pre- and Post-Optimization Results**

The ensemble models performed significantly better according to the improvements shown in various metrics, and some clear practical benefits have been derived relating to mobile security.

Before optimization, the models were giving quite a good performance with 91.37% accuracy for DT, whereas RF and GBM were performing equally well to give an accuracy of 87.63%. After optimization, these values increase to 99.70% for DT, 98.88% for RF, and 99.18% for the GBM. Thereby, these improvements are staying very important in real mobile security where high accuracy is of utmost need to minimize false positives and false negatives at the time of threat detection. Classes pre- and post-optimization are quite spiky: pre-optimization precision, recall, F1-scores presented inconsistent results; specifically, Class 2, which represents the complex attack vectors, has almost 0%

for the precision score. The same tests present much-improved results after optimization to over 99% precision for all models and classes, with major lift-ups in both recall and F1-scores. This strongly reduces missed detections and false alarms, one of the key factors toward solid real-world security.

This optimization also improved convergence speed by reducing the training time by 25%. It was done through tuning critical parameters, learning rate, and number of trees, as shared in Figure 3. The faster convergence essentially means quicker updates for dynamic environments, ensuring that any emerging threats are addressed with urgency. Pre-optimization, models used to overfit, limiting their generalization to unseen data. The subsequent regulation adjustments, coupled with the adjustments in the tree depth, helped in introducing robustness such that the resultant models, upon optimizations done on them, could handle new security threats with increased aplomb. This is important in mobile security, where any models are supposed to adapt to evolving attack vectors. These improvements, especially the huge jumps in accuracy, precision, recall, and speed, have made the optimized models more effective for real-world deployment in mobile security systems. This would make for higher detection capability, faster training times, and thus keep mobile applications secure against threats-both known and emerging-with less computational overhead.

6 Discussion

The broad and multi-faceted nature of ensemble learning integrated with meta-heuristics in the context of multiclass classification encompasses an entire bundle of complications and nuances. Hence, our study "Optimal Ensemble Learning with Meta-Heuristics for Multi-Class Classification of Syscall Binder Interactions in Mobile Applications" was intended to enlighten into such complexities. Further, this section will deliberate upon the examination and analysis of the results found, obstacles encountered, and the importance of our research.

All ensemble learning methods have been extended to other domains, such as stock price prediction. Nabi et al., (2020) developed a new approach to multiclass classification ensemble learning for stock price prediction in this respect. The proposed method has been developed and tested using historical data. Ensemble learning has been applied to identify several mental disorders in mental health, and (Nasrullah & Jalali, (2022) have proposed a deep learning model using ensemble techniques for this purpose. Furthermore, meta-learning methods can enhance optimization even more at the ensemble step when dealing with multiclass classification. Vogiatzis et al., (2022) introduced new meta-learning methods for the multiclass image classification problem. Their objective was to improve the performance of ensemble learning by considering the optimization of combining different models. Our results are consistent with some recent works on metaheuristics in machine learning. For instance, both (Nanda et al., 2022; Wakjira et al., 2022) improved prediction accuracy by integrating optimization techniques and machine learning models in the healthcare and engineering domains, respectively. Similarly, the work of Saminathan & Thangavel depicted the effectiveness of nature-inspired algorithms such as the Fruit Fly Optimization Algorithm in mobile networking contexts. We current some advancements from prior art by displacing these approaches within a mobile security context, in which hyper-metaheuristics was used to optimize classification in syscall-binder interactions. The novelty of its application in this context not only improves accuracy within detection but also provides evidence of the flexibility of some optimization techniques in enhancing complex classification tasks in security-focused domains.

Combining ensemble learning with meta-heuristics possibly comprises one of the fruitful approaches toward handling multiclass classification problems. Ensemble learning has great potential to substantially enhance the performance and robustness of multiclass classification systems in a variety of domains by effectively using the collective capabilities provided by multiple models.

7 Conclusion and Future Work

The outcome of our research proved greatly informational and enlightening, providing an in-depth insight into multiclass categorization in the domain of mobile applications. The most significant contribution of our study was the presence and impact brought in by adding meta-heuristics into the traditional models of ensemble learning. The base learners' performance metrics, the DT, RF, and GBM, received a giant boost once optimized with meta-heuristics. This was notably evident in categorizing the courses, causing difficulties in their earlier categorization. Besides enhancing all the different models, the meta-heuristics heuristic refined the ensemble performance comparable to its constituent models, if not better. This case exemplifies a complex classification task where combining ensemble learning with meta-heuristics makes it possible to attain synergy between them. Classification of Syscall-Binder interactions lie at the juncture of functionality and security regarding mobile phone applications. Our research has significant implications in order to enhance the security of mobile applications. Classification with precision takes away much ambiguity and can even open ways to discover potential weaknesses and aberrant behaviors more accurately. Combined use of ensemble learning and meta-heuristics provides enlarged capabilities in classification, thereby enhancing the possibility of prompt detection and mitigation of harmful activities or inappropriate syscall-binder interactions. This enhances the security posture of individual applications and bolsters the general security ecosystem of mobile platforms.

Although this research opened one avenue, the area of ensemble learning is so diverse when combined with meta-heuristic, which offers many other opportunities to explore. Future research could be focused on applying further nature-inspired optimization methods, extending the portfolio of meta-heuristic algorithms beyond the ones used thus far, namely, Fruit Fly Optimization and Firefly Algorithm. What is most interesting for further research is the possible applicability of our approach to a wide range of domains other than the Syscall-Binder interactions of mobile applications. Another direction of possible improvement can be done by integrating deep learning techniques with meta-heuristic algorithms, which could bestow a new phase of categorization skills. Furthermore, the immediate application of our methodology, which ensures constantly updated classification in the changing contexts of the mobile world, acts as a base for the development of future research activities.

It follows that hyper-metaheuristic optimization significantly enhances the performance of the ensemble learning models at classifying syscall-binder interactions, hence making them highly applicable in real-world mobile security. These enhanced detection capabilities can then be integrated into a mobile security system to determine threats such as malware and unauthorized data access with minimized false positives and negatives. What is more, faster convergence and improved generalization enable real-time adaptability to emerging threats, which is very important in dynamic mobile environments. Besides, metaheuristics extensions might also be performed beyond Fruit Fly Optimization and Firefly Algorithm for even better performance. For example, one could investigate using Genetic Algorithms, Particle Swarm Optimization, or even hybridized metaheuristics to achieve better performance and adaptability. On the other hand, such an algorithm's testing, combined with other machine learning methods, can yield even better results while detecting newly emerging sophisticated threats in mobile ecosystems. This significantly improved the current research on "Optimal Ensemble Learning with Meta-heuristics for Multiclass Classification of Syscall-Binder Interactions in Mobile

Applications." Yet, the domain of possibility remains great, luring further research, development, and enhancement. The incorporation of ensemble learning and meta-heuristics opens an exciting future direction, which may inspire future researchers and point out new possibilities or novel frameworks not taken before.

References

[1] AlKhereibi, A. H., Wakjira, T. G., Kucukvar, M., & Onat, N. C. (2023). Predictive machine learning algorithms for metro ridership based on urban land use policies in support of transit-oriented development. *Sustainability*, *15*(2), 1718. https://doi.org/10.3390/su15021718

[2] Ammi, M., & Jama, Y. M. (2023). Cyber Threat Hunting Case Study using MISP. *Journal of Internet Services and Information Security*, *13*(2), 1-29. https://doi.org/10.58346/JISIS.2023.I2.001

[3] Anand, U., & Shrivastava, V. (2024). Digital Leadership: Exploring the Role of Top Management in Digital Transformation. *Global Perspectives in Management*, 2(2), 1-11.

[4] Breiman, L. (1996). Bagging predictors. *Machine learning*, *24*, 123-140. https://doi.org/10.1007/BF00058655

[5] Bühlmann, P., & Hothorn, T. (2007). Boosting algorithms: Regularization, prediction and model fitting. https://doi.org/10.1214/07-STS242

[6] Cao, B., Zhong, W., Xie, X., Zhang, L., & Qing, Y. (2022). A Multi-modal Feature Fusion-based Approach for Mobile Application Classification and Recommendation. *Journal of Internet Technology*, *23*(6), 1417-1427. https://doi.org/10.53106/160792642022112306023

[7] Cavus, N., & Ibrahim, D. (2017). Learning English using children's stories in mobile devices. *British Journal of Educational Technology*, *48*(2), 625-641. https://doi.org/10.1111/bjet.12427

[8] Cheng, J., & Shi, T. (2022). Structural optimization of transmission line tower based on improved fruit fly optimization algorithm. *Computers and Electrical Engineering*, *103*, 108320. https://doi.org/10.1016/j.compeleceng.2022.108320

[9] D'Ancona, G., Massussi, M., Savardi, M., Signoroni, A., Di Bacco, L., Farina, D., ... & Benussi, S. (2023). Deep learning to detect significant coronary artery disease from plain chest radiographs AI4CAD. *International Journal of Cardiology*, *370*, 435-441. https://doi.org/10.1016/j.ijcard.2022.10.154

[10] Dasari, A. K., Biswas, S. K., Thounaojam, D. M., Devi, D., & Purkayastha, B. (2023, March). Ensemble Learning Techniques and Their Applications: An Overview. In *International Conference on Communications and Cyber Physical Engineering 2018* (pp. 897-912). Singapore: Springer Nature Singapore. https://doi.org/10.1007/978-981-19-8086-2_85

[11] Dinh, H. T., Lee, C., Niyato, D., & Wang, P. (2013). A survey of mobile cloud computing: architecture, applications, and approaches. *Wireless communications and mobile computing*, *13*(18), 1587-1611. https://doi.org/10.1002/wcm.1203

[12] Faruki, P., Bharmal, A., Laxmi, V., Ganmoor, V., Gaur, M. S., Conti, M., & Rajarajan, M. (2014). Android security: a survey of issues, malware penetration, and defenses. *IEEE communications surveys & tutorials*, *17*(2), 998-1022. https://doi.org/10.1109/COMST.2014.2386139

[13] Feng, W., Gou, J., Fan, Z., & Chen, X. (2023). An ensemble machine learning approach for classification tasks using feature generation. *Connection Science*, *35*(1), 2231168. https://doi.org/10.1080/09540091.2023.2231168

[14] Ghazi, A., Aljunid, S. A., Idrus, S. Z. S., Endut, R., Rashidi, C. B. M., Ali, N., ... & Sharma, T. (2021, February). Hybrid WDM and Optical-CDMA over Multi-Mode Fiber Transmission System based on Optical Vortex. In *Journal of Physics: Conference Series* (Vol. 1755, No. 1, p. 012001). IOP Publishing. https://doi.org/10.1088/1742-6596/1755/1/012001

[15] Javed, F., Afzal, M. K., Sharif, M., & Kim, B. S. (2018). Internet of Things (IoT) operating systems support, networking technologies, applications, and challenges: A comparative review. *IEEE Communications Surveys & Tutorials*, *20*(3), 2062-2100. https://doi.org/10.1109/COMST.2018.2817685

[16] Kappagantula, S., & Mannayee, G. (2024). Dynamic Path Planning Algorithm for Mobile Robots: Leveraging Reinforcement Learning for Efficient Navigation. *Journal of Internet Services and Information Security, 14*(2), 226-236. https://doi.org/10.58346/JISIS.2024.I2.014

[17] Karimov, N., Kalandarova, M., Makhkamova, N., Asrorova, Z., Saydamatov, F., Ablyakimova, R., Karshiboeva, M., & Odilov, B. (2024). Impact of Mobile Applications on Tourism Development in Uzbekistan. *Indian Journal of Information Sources and Services*, *14*(4), 175–181. https://doi.org/10.51983/ijiss-2024.14.4.27

[18] Kausar, N., Hameed, A., Sattar, M., Ashraf, R., Imran, A. S., Abidin, M. Z. U., & Ali, A. (2021). Multiclass skin cancer classification using ensemble of fine-tuned deep learning models. *Applied Sciences*, *11*(22), 10593. https://doi.org/10.3390/app112210593

[19] Kim, K., Ko, E., Kim, J., & Yi, J. H. (2019). Intelligent Malware Detection Based on Hybrid Learning of API and ACG on Android. *Journal of Internet Services and Information Security*, *9*(4), 39-48.

[20] Kitana, A., Traore, I., & Woungang, I. (2020). Towards an Epidemic SMS-based Cellular Botnet. *Journal of Internet Services and Information Security*, *10*(4), 38-58. https://doi.org/10.22667/JISIS.2020.11.30.038

[21] Kumawat, B. (2012). A research study on packet forwarding attacks in mobile ad-hoc networks. *International Journal of Communication and Computer Technologies, 1*(1), 34-38. https://doi.org/10.31838/IJCCTS/01.01.04

[22] Lee, H., Kim, C., Kim, H., Kim, J., & Kang, M. (2023). Improving Active Collaborative Clinical Learning Through a Mobile Application for Undergraduate Nursing Students. *CIN: Computers, Informatics, Nursing*, *41*(5), 309-315. https://doi.org/10.1097/CIN.0000000000000918

[23] Lee, S., Nguyen, N. H., Karamanli, A., Lee, J., & Vo, T. P. (2023). Super learner machine-learning algorithms for compressive strength prediction of high performance concrete. *Structural Concrete*, *24*(2), 2208-2228. https://doi.org/10.1002/suco.202200424

[24] Lestari, I., Maksum, A., & Kustandi, C. (2019). Mobile Learning Design Models for State University of Jakarta, Indonesia. *International Journal of Interactive Mobile Technologies (iJIM)*, *13*(09), 152. https://doi.org/10.3991/ijim.v13i09.10987

[25] Liu, C., & Correia, A. P. (2021). A Case Study of Learners' Engagement in Mobile Learning Applications. *Online Learning*, *25*(4), 25-48. https://doi.org/10.24059/olj.v25i4.2827

[26] Luh, R., Janicke, H., & Schrittwieser, S. (2019). AIDIS: Detecting and classifying anomalous behavior in ubiquitous kernel processes. *Computers & Security*, *84*, 120-147. https://doi.org/10.1016/j.cose.2019.03.015

[27] Mahdavifar, S., Alhadidi, D., & Ghorbani, A. A. (2022). Effective and efficient hybrid android malware classification using pseudo-label stacked auto-encoder. *Journal of network and systems management*, *30*(1), 22. https://doi.org/10.1007/s10922-021-09634-4

[28] Mahdavifar, S., Kadir, A. F. A., Fatemi, R., Alhadidi, D., & Ghorbani, A. A. (2020, August). Dynamic android malware category classification using semi-supervised deep learning. In *2020 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech)* (pp. 515-522). IEEE. https://doi.org/10.1109/DASC-PICom-CBDCom-CyberSciTech49142.2020.00094

[29] Mekdad, Y., Aris, A., Babun, L., El Fergougui, A., Conti, M., Lazzeretti, R., & Uluagac, A. S. (2023). A survey on security and privacy issues of UAVs. *Computer Networks*, *224*, 109626. https://doi.org/10.1016/j.comnet.2023.109626

[30] Mishra, S., Shaw, K., Mishra, D., Patil, S., Kotecha, K., Kumar, S., & Bajaj, S. (2022). Improving the accuracy of ensemble machine learning classification models using a novel bit-fusion algorithm for healthcare AI systems. *Frontiers in Public Health*, *10*, 858282. https://doi.org/10.3389/fpubh.2022.858282

[31] Mohammed, A., & Kora, R. (2023). A comprehensive review on ensemble deep learning: Opportunities and challenges. *Journal of King Saud University-Computer and Information Sciences*, *35*(2), 757-774. https://doi.org/10.1016/j.jksuci.2023.01.014

[32] Monisha, S., Monisha, M., Deepa, P., Sathya, R., & Gunasekaran, K. (2019). An android application for exhibiting Statistical chronicle information. *International Journal of Communication and Computer Technologies*, *7*(1), 7-9.

[33] Nabi, R. M., Soran Ab M, S., & Harron, H. (2020). A novel approach for stock price prediction using gradient boosting machine with feature engineering (gbm-wfe). *Kurdistan Journal of Applied Research*, *5*(1), 28-48. https://doi.org/10.24017/science.2020.1.3

[34] Nanda, R., Nath, A., Patel, S., & Mohapatra, E. (2022). Machine learning algorithm to evaluate risk factors of diabetic foot ulcers and its severity. *Medical & biological engineering & computing*, *60*(8), 2349-2357. https://doi.org/10.1007/s11517-022-02617-w

[35] Nascimento, D. S. C., Canuto, A. M. P., & Coelho, A. L. V. (2016). Multi-label meta-learning approach for the automatic configuration of classifier ensembles. *Electronics Letters*, *52*(20), 1688-1690. https://doi.org/10.1049/el.2016.1446

[36] Nasrullah, S., & Jalali, A. (2022). [Retracted] Detection of Types of Mental Illness through the Social Network Using Ensembled Deep Learning Model. *Computational Intelligence and Neuroscience*, *2022*(1), 9404242. https://doi.org/10.1155/2022/9404242

[37] Ndirangu, D., Mwangi, W., & Nderu, L. (2019). Using ensemble technique to improve multiclass classification. *Journal of Information Engineering and Applications*, *9*(5), 28-42. https://doi.org/10.7176/JIEA/9-5-04

[38] Ojaghloo, M., & Jannesary, A. (2015). Investigate all attacks on Mobile Wireless Networks and Finding security solutions. *International Academic Journal of Innovative Research*, *2*(2), 17-27

[39] Olisah, C. C., Smith, L., & Smith, M. (2022). Diabetes mellitus prediction and diagnosis from a data preprocessing and machine learning perspective. *Computer Methods and Programs in Biomedicine*, *220*, 106773. https://doi.org/10.1016/j.cmpb.2022.106773

[40] Pasha, S. A., Karimulla, S., Komalvenkatsai, P., Anil, P., & Venu, M. (2021). Integrated Mobile Veld Fire Detection and Alerting System for Rural Communities. *International Journal of communication and computer Technologies*, *9*(2), 37-42.

[41] Sagi, O., & Rokach, L. (2018). Ensemble learning: A survey. *Wiley interdisciplinary reviews: data mining and knowledge discovery*, *8*(4), e1249. https://doi.org/10.1002/widm.1249

[42] Sainin, M. S., & Alfred, R. (2012, September). A direct ensemble classifier for imbalanced multiclass learning. In *2012 4th Conference on Data Mining and Optimization (DMO)* (pp. 59-66). IEEE. https://doi.org/10.1109/DMO.2012.6329799

[43] Saminathan, K., & Thangavel, R. (2022). Energy efficient and delay aware clustering in mobile adhoc network: A hybrid fruit fly optimization algorithm and whale optimization algorithm approach. *Concurrency and Computation: Practice and Experience*, *34*(11), e6867. https://doi.org/10.1002/cpe.6867

[44] Saxena, N., Hayes, E., Bertino, E., Ojo, P., Choo, K. K. R., & Burnap, P. (2020). Impact and key challenges of insider threats on organizations and critical businesses. *Electronics*, *9*(9), 1460. https://doi.org/10.3390/electronics9091460

[45] Shahhosseini, M., Hu, G., & Pham, H. (2022). Optimizing ensemble weights and hyperparameters of machine learning models for regression problems. *Machine Learning with Applications*, *7*, 100251. https://doi.org/10.1016/j.mlwa.2022.100251

[46] Sihag, V., Prakash, S., Choudhary, G., Dragoni, N., & You, I. (2022, November). DIMDA: deep learning and image-based malware detection for Android. In *Futuristic Trends in Networks and*

Computing Technologies: Select Proceedings of Fourth International Conference on FTNCT 2021 (pp. 895-906). Singapore: Springer Nature Singapore. https://doi.org/10.1007/978-981-19-5037-7_64

[47] Sivakumar, K., Banerjee, K., Vibin, R., Saravanan, B., Srivastava, S., Anand, R., & Bhoopathy, V. (2024). AI-driven green network management for future Internet and software-defined networking (SDN). *Journal of Environmental Protection and Ecology, 25*(6), 2133–2144.

[48] Vaithiyanathan, V., Mani, V., Govindasamy, S., & Selvaraj, J. (2024). A Multi-Objective Time–Series Optimization for Optimum Planning Design of Integrative Power System with the Effects of Multi-Dimensional Sources of Uncertainty. *Electric Power Components and Systems*, *52*(6), 891-904. https://doi.org/10.1080/15325008.2023.2237017

[49] Vogiatzis, A., Orfanoudakis, S., Chalkiadakis, G., Moirogiorgou, K., & Zervakis, M. (2022). Novel Meta-Learning Techniques for the Multiclass Image Classification Problem. *Sensors*, *23*(1), 9. https://doi.org/10.3390/s23010009

[50] Wakjira, T. G., Abushanab, A., Ebead, U., & Alnahhal, W. (2022). FAI: Fast, accurate, and intelligent approach and prediction tool for flexural capacity of FRP-RC beams based on super-learner machine learning model. *Materials Today Communications*, *33*, 104461. https://doi.org/10.1016/j.mtcomm.2022.104461

[51] Zhang, Q., Li, C., Yin, C., Zhang, H., & Su, F. (2022). A hybrid framework model based on wavelet neural network with improved fruit fly optimization algorithm for traffic flow prediction. *Symmetry*, *14*(7), 1333. https://doi.org/10.3390/sym14071333

[52] Zhang, Y., Yang, M., Yang, Z., Gu, G., Ning, P., & Zang, B. (2014). Permission use analysis for vetting undesirable behaviors in android apps. *IEEE transactions on information forensics and security*, *9*(11), 1828-1842. https://doi.org/10.1109/TIFS.2014.2347206

Authors Biography

Dr. Mohammad Othman Nassar is an Associate Professor of Computer Information System in Amman Arab University; he received his Ph.D. in Computer Information System in 2009. Dr. Nassar has 15 years' teaching experience. He has been working at Amman Arab University since 2010. His is the director of marketing department in Amman Arab University. He occupied different leading positions at Amman Arab University: head for Computer Information System department, eLearning center Manager, and computer center manager. He has published more than 40 scientific research publications in international peer reviewed journals and conferences. Dr. Nassar is an editorial board member for 4 international peer reviewed journals.

Dr. Feras Fares Al-Mashagba is an Associate Professor of Artificial Intelligence in Jarash University; he received his Ph.D. in Computer Information System/ Artificial Intelligence in 2009. Dr. Almashakbah has 15 years' teaching experience. He has published 30 scientific research publications in international peer reviewed journals and conferences. Dr. Almashakbah is an editorial board member for 3 international peer reviewed journals.