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

# Intrusion Detection Using an Improved Cuckoo Search Optimization Algorithm

Mutasem K. Alsmadi[1], Dr. Rami Mustafa A Mohammad[2*], Malek Alzaqebah[3], Sana Jawarneh[4], Muath AlShaikh[5], Ahmad Al Smadi[6], Fahad A. Alghamdi[7], Jehad Saad Alqurni[8], and Hayat Alfagham[9]

[1]Department of MIS, College of Applied Studies and Community Service, Imam Abdulrahman Bin Faisal University, City of Dammam, Saudi Arabia. mksalsmadi@gmail.com, https://orcid.org/0000-0001-6892-8399

[2*]Saudi Aramco Cybersecurity Chair, Department of Computer Information Systems, College of Computer Science and Information Technology, Imam Abdulrahman Bin Faisal University, Dammam, Saudi Arabia. rmmohammad@iau.edu.sa, https://orcid.org/0000-0002-2612-1615

[3]Department of Mathematics, College of Science, Imam Abdulrahman Bin Faisal University, City of Dammam, Saudi Arabia; Basic & Applied Scientific Research Center, Imam Abdulrahman Bin Faisal University, Dammam, Saudi Arabia. maafehaid@iau.edu.sa, https://orcid.org/0000-0002-3846-0673

[4]Department of Computer Science, Community College, Imam Abdulrahman Bin Faisal University, City of Dammam, Saudi Arabia. sijawarneh@iau.edu.sa, https://orcid.org/0000-0002-9863-3775

[5]Department of Computer Science, College of Computing and Informatics, Saudi Electronic University, Riyadh, KSA, Saudi Arabia. m.alshaikh@seu.edu.sa, https://orcid.org/0000-0001-5550-7659

[6]Department of Data Science and Artificial Intelligence, Zarqa University, Zarqa, Jordan. aalsmadi@zu.edu.jo, https://orcid.org/0000-0003-3487-8041

[7]Department of MIS, College of Applied Studies and Community Service, Imam Abdulrahman Bin Faisal University, City of Dammam, Saudi Arabia. faghamdi@iau.edu.sa, https://orcid.org/0000-0003-1996-9113

[8]Department of Educational Technologies, College of Education, Imam Abdulrahman Bin Faisal University, Dammam, Saudi Arabia. jalqurni@iau.edu.sa, https://orcid.org/0000-0002-4834-9039

[9]Department of MIS, College of Applied Studies and Community Service, Imam Abdulrahman Bin Faisal University, City of Dammam, Saudi Arabia. hmalfagham@iau.edu.sa, https://orcid.org/0000-0003-2815-1049

Received: January 28, 2024; Revised: February 27, 2024; Accepted: April 16, 2024; Published: June 29, 2024

*Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications (JoWUA)*, volume: 15, number: 2 (June), pp. 73-93. DOI: *10.58346/JOWUA.2024.I2.006*

*Corresponding author: Saudi Aramco Cybersecurity Chair, Department of Computer Information Systems, College of Computer Science and Information Technology, Imam Abdulrahman Bin Faisal University, Dammam, Saudi Arabia.

## Abstract

These days, intelligent cybersecurity models based on machine learning and data mining techniques are prevalent. Several factors might affect the quality of these models, including the accuracy, the ability to train new models quickly, the quick decision-making process, the simplicity of the created models, and the model's interpretability. Feature selection algorithms can help achieve all these characteristics by isolating the crucial features from the unimportant ones during the model creation phase. The current article proposes an intelligent intrusion detection model based on an improved cuckoo search algorithm which is a nature-inspired optimization algorithm. The improved cuckoo search algorithm proposed in this article may tolerate several bad steps toward determining the set of effective features that would preserve or maximize the capabilities of the produced classification models. The generalization ability of such an algorithm is examined by applying it to 10 benchmark datasets, and it showed superior outcomes compared with several nature-inspired attribute selection approaches. Later, the improved cuckoo search algorithm is used to develop intelligent intrusion detection systems using the well-known "NSL-KDD" dataset. The obtained outcomes are appealing regarding the general performance, the time required to develop the intelligent intrusion detection models, and the number of rules generated.

**Keywords:** Intrusion Detection, Anomaly Detection, Cuckoo Algorithm, Great Deluge, Feature Selection, NSL-KDD.

# 1 Introduction

Cyber security is often identified as various strategies and approaches designed to secure critical information, applications, computers, and networks from devastation, modification, illegal use, and other cybercrimes. Cyber security systems might be computer-based systems or network-based systems. All these systems typically comprise antivirus tools, firewalls, and Intrusion Detection Systems (IDS). IDSs aid in revealing illegitimate usage, illegal modification, and deliberate data devastation (Sreenivasu et al., 2023; Nikitina et al., 2023).

Generally, there exist three types of IDSs:

- Anomaly detection systems may identify malevolent activities by tracking the difference from the normal network's traffic. These systems may realize emergent abnormalities that are usually called zero-day attacks. A possible benefit of anomaly-based approaches is that activity patterns are adjusted for each networking system. It is, therefore, difficult for cybercriminals to comprehend what strategies they might perform without getting spotted. Additionally, the details accumulated from anomaly detection systems may be utilized to produce signatures and rules for construction misuse recognition systems. Nevertheless, these systems sometimes face high false alerts because formerly unobserved genuine activities might occasionally be classified as anomalies.
- Abuse recognition systems may identify known harmful attempts depending on the revealed patterns called signatures. Nevertheless, this type involves a constant and quick improvement of the signatures and rules for detecting unique and emerging attempts.
- Hybrid systems integrate anomaly and abuse systems. They are often utilized to improve detection rates and reduce the false positive ratios for emergent vulnerabilities. Most existing intrusion detection systems tend to be hybrid systems. That is why Machine Learning (ML) and Data Mining (DM) based and hybrid and anomaly systems were illustrated alongside (Yang et al., 2022).

A variety of techniques have been utilized for constructing IDSs. Intelligent models dependent on DM and ML proved reliable techniques for revealing intrusions (Elshrkawey et al., 2021). Generally, the learning strategies in DM and ML could be either supervised or unsupervised. The principal difference between them is that supervised learning suggests that there is a previous insight into what exactly the class value for each dataset instance is. Consequently, the primary purpose of supervised training is to build models that when supplied with a test dataset, will ideally approximate the relation between input features and a class variable. Conversely, unsupervised learning normally does not include a class variable. So, it is designed to infer the natural structure within the collected dataset items.

Classification is a DM technique that assigns instances within a dataset to a class or a target variable. Intrusion detection belongs to the supervised classification training strategy. Different intelligent methods were used for developing IDS systems, including Support Vector Machine (SVM) (Mohammed & Sulaiman, 2012; Mohammad, 2022), Neural Networks (NN) (Shenfield et al., 2018; Mohammad, 2018), Fuzzy Logic (FL) (Mkuzangwe & Nelwamondo, 2017), Hybrid Intelligent System (HIS) (Bhumgara & Pitale, 2019), and Decision Tree (DT) (Zhang et al., 2018; Mohammad & Alqahtani, 2019). Nonetheless, among the obstacles of current campaigns is the low True Positive (TP) and/or the high False Positive (FP) ratios (Srinivasareddy et al., 2021).

Generally speaking, a feasible explanation behind creating imperfect intelligent IDSs might be the large training dataset employed within the training stage. In its initial shape, a training dataset will probably comprise some extra attributes that complex the produced classification model without providing an apparent good effect on the model's performance (Babenko et al., 2021). Some feature selection techniques might be incorporated with DM techniques to decrease the training dataset's dimensionality. This might even decrease the time required for training the models (Yağız et al., 2022). Besides, not using a proper feature selection strategy may create complicated models that are hard to comprehend. Therefore, choosing the right group of attributes, as well as reducing the training dataset dimensionality, is a vital preprocessing stage for creating DM models. Attribute selection techniques are divided into filter, wrapper, and embedded methods. Embedded strategy determines the best attributes during the training stage (Liu et al., 2019; Salim et al., 2023). Common embedded strategies employ regularization approaches to penalize trivial attributes. Wrapper strategies select the best attributes by assessing every set of attributes and evaluating their detailed effect regarding the class attribute based on a specific DM technique. So, this method involves utilizing a DM algorithm to generate models during the attribute selection process for each viable group of attributes (Asl et al., 2022).

Moreover, it entails assessing all possible combinations of attributes in the training dataset to obtain the attributes that might generate the best models per the final results of the DM algorithm. For instance, assume a dataset that has *"x"* attributes. Thus, there will be 2x-1 possible combinations of attributes that the classification algorithm will assess during the attribute selection process. Later, the attributes that produce the classification model that attained the top performance are selected.

Nevertheless, such a strategy is computationally costly, and experts constantly look for methods for cost-effectively selecting the best set of attributes. Filter strategies use different measurements that primarily rely on *"Information Theory" (Cover, 1999)* and several statistics measurements to realize how strong the relationship is between the input attributes and the class attribute (Mohammad, 2020) and might be used for ranking the attributes and picking the best group according to a predefined selection strategy. Additionally, such techniques may be applied in the preprocessing stage with other methods, such as Correlation Feature Set, and Minimum Redundancy Maximum Relevance, to obtain the best group of attributes (Mohammad, 2020). Filter strategies outperformed several different selection strategies depending on various conditions. Filter strategies tend to be computationally better than wrapper and embedded techniques.

Moreover, filter strategies might be less influenced by any DM algorithm and robust to several issues associated with overfitting considering that filter strategies select a group of important attributes while not using any DM algorithm. In reality, most IDSs are afflicted by the so-named drift of concept (Mohammad et al., 2014; Mohammad, 2020), in which the group of attributes for creating IDSs keeps altering with time. For example, the group of important features at time slot S1 could convert ineffective at time slot S2, and the ineffective ones at S1 become significant at time S2. Thus, selecting a group of attributes is crucial before creating IDSs. However, the attribute selection process should improve performance, enhance simplicity, and advance the created models' robustness.

The current research introduces an Improved Cuckoo Search algorithm for feature selection. Cuckoo algorithm CA is inspired by the interesting cuckoo bird's reproduction strategy (Yang & Deb, 2009). CA has several benefits compared to different nature-inspired techniques, including the Particle swarm algorithm (Abualigah et al., 2018), Ant colony algorithm (Rais & Mehmood, 2018), Cat swarm algorithm (Lin et al., 2016), and Grey wolf algorithm (Shen, & Zhang, 2022), given that it is looking at some elitism. In addition, in CA, entropy is more beneficial to define the moving size, which is heavily tailored to a probably big moving size. Moreover, since fewer parameters need to be adjusted, it might be easy to accommodate a wider variety of optimization problems (Alzaqebah et al., 2021). However, one of the CA's main issues is its slower convergence speed (Alzaqebah et al., 2021; Saadawi et al., 2024; Prasad Babu et al., 2023). An improved CA using rough sets has been presented in (Aziz & Hassanien, 2018), where many cuckoo species use the parasitic position of the obligatory brood, and various birds use Lévy flight behaviors. The current article suggests an improved CA that addresses the slow convergence of CA and prevents the CA from getting caught in local optima. Overall, the main pros of the presented approach are:

- It improves the required computation, making it more efficient.
- It reduces the training dataset dimensionality.
- It can help in creating easy-to-interpret IDSs.

## 2 Background and Previous Studies

Different DM and ML models were suggested to reveal intrusion attacks. This section goes over these models to understand their success and the possible opportunities for enhancement.

In (Balakrishnan et al., 2014), a novel attribute selection method named *"Optimal Feature Selection algorithm based on information gain Ratio"* has been advised and utilized to reveal intrusions. This algorithm chooses the best attributes within the popular dataset in the *"KDD Cup" (Rosset & Inger, 2000)*. Two DM algorithms have been utilized in creating the IDS systems, mainly *"Rule-Based"* and *"Support Vector Machine."* The empirical findings indicated that the produced IDSs efficiently decreased the false alerts and discovered a certain attack more accurately, which is the *"Denial of Service Attacks.".* In (Mohammad & Alsmadi, 2021), a creative attributes selection technique was offered and is named *"the Highest Wins"* (HW). The HW was easy and simple to comprehend. It finds the variance between the real and the anticipated likelihood values. The experimental outcomes revealed that the suggested technique created ambitious results regarding various evaluation metrics.

In (Abdullah et al., 2018), the article proposed a method that splits the training data into diverse subsets according to the kind of attacks. Then attribute selection was achieved through the *"Information Gain"* algorithm for each subset. The best attributes were produced by combining the attributes attained for every attack. Empirical analysis completed on *"NSL-KDD"* indicated that the suggested technique improved the precision ratio and moderated the complexity of the produced models. The method developed in (Pandey, 2019), begins by selecting the attributes by using *"Information Gain"* which is linked with the base algorithm to select the best group of attributes to build voting-based IDSs. Various DM algorithms were utilized for creating the voting model, including *"Naïve Bayes"*, *"Decision Tree"*, *"AdaBoost"*, and *"Random Forest"*. In (Kumar & Batth, 2016), the researchers employed three traditional methods for attribute selection. This study assessed the capability of the proposed technique using three DM techniques, namely *"REPTree"*, *"Naive Bayes"* and *"Decision Tree"*. The 10-fold cross-validation was employed in comparing different performance measures. The *"NSL-KDD"* dataset was utilized for building the models. The outcomes indicated that the improved *"Naive Bayes"* provided a better precision ratio and decreased false alerts.

In (Kannan et al., 2015), the researchers suggested a creative cloud IDS to identify the attackers within a hybrid virtualized cloud platform. Furthermore, a unique attribute selection method called *"Temporal Constraint based on Feature Selection algorithm"* was suggested. Additionally, the authors advised an innovative classification technique called *"Hybrid Decision Tree"*. This method is an extension of the original decision tree algorithm. The empirical outcomes indicated that the recommended method advanced the detecting accuracy and minimized the false alerts. In the project carried out in (Othman et al., 2018), the *"ChiSqSelector"* (Karim & Kaysar, 2016) is employed for picking out the most beneficial group of attributes to construct a *"Support Vector Machine"* based IDSs using *"Apache Spark Big Data Platform"*. The *"KDD-99" (Rosset & Inger, 2000)* was employed for building the system. Empirical outcomes indicated that the *"Chi-SVM"* surpassed *"Chi-Logistic Regression"* with regard to attained accuracy and the time required for building the models. Ensemble DM strategy has also coupled with attribute selection strategies to generate IDS. For instance, the effort in (Almasoudy et al., 2020) recommended an innovative " CFS-BA " algorithm to decrease the dataset dimensionality. This technique relies on the association among the attributes for choosing the best group of attributes. Next, three DM techniques were utilized for creating the ensemble IDS system, those were *"Forest by Penalizing Attributes"*, *"Random Forest"*, and *"Decision Tree"*. The datasets employed for assessing the proposed techniques were *"AWID", "CIC-IDS2017",* and *"NSL-KDD"* where the outcomes indicated that the proposed technique provided improved results in comparison to various methods considering different assessment measures. The wrapper strategy was utilized to choose the best attribute for revealing intrusions. The technique offered in (Almasoudy et al., 2020), which employed a wrapper method based on *"Differential Evolution"* (DE), is a good example of these techniques. Once the best group of attributes using DE is defined, the *"Extreme Learning Machine"* creates the detection system. DE continues operating until the best accuracies are attained. The proposed method proved efficient, specifically in binary and five-based attacks (Camgözlü et al., 2023).

The project in (Ambusaidi et al., 2014) is an additional case study of the models that employed the wrapper method for selecting the most crucial attributes for developing IDSs. In this research, the wrapper attributes selection strategy was coupled with the hybrid attribute selection strategy. Two major phases are comprised in this method. The upper level performs a preliminary effort to obtain the best group of attributes, in which the mutual info between the class attribute and the input attributes works as a determinant measurement. The picked group of attributes generated in this phase are refined in the lower level using the wrapper strategy in which the *"Least Square Support Vector Machine"* (*LSSVM*) was employed for selecting the attributes. Experimental assessments revealed that the proposed system is very effective in revealing intrusions. An inspiring algorithm called *"Maximum Dependence Maximum Significance Algorithm"* was offered in (Senthilnayaki et al., 2019) for choosing the best group of attributes to create an IDS using *"K-neighbors"*.

In (Bhaskar et al., 2019), the *"Adaptive Jaya Method"* was utilized to find the group of reliable attributes to generate IDSs that reduce the false alert ratio and increase the detecting accuracy ratio. In (Shah et al., 2017), IDS was developed by using *"Sparse Logistic Regression"* (SPLR). SPLR is an innovative technique utilized for attribute selection using an algorithm that selects a modest group of attributes from the original dataset to create classification systems. Genetic Algorithms (GA) were utilized in (Ren et al., 2019) for choosing the reduced group of attributes for constructing IDSs. The method begins with an *"Isolation Forest"* to eliminate the outliers of the dataset, and then GA was employed for producing the collection of reliable attributes. The system is contrasted with other systems, revealing ambitious results regarding several assessment criteria.

In general, the attributes selection strategy is a crucial preprocessing stage for creating accurate intelligent IDSs with minimum false alerts. Yet, the better the attribute selection technique, the more successful the IDS. The present research will develop an improved attribute selection technique to build intelligent IDS systems capable of correctly recognizing intrusion attempts. The proposed attribute selection algorithm combines two nature-inspired approaches, which, to our knowledge, have never been combined once before and applied for building IDSs. These algorithms are the *"Cuckoo Search Algorithm"* CSA and the *"Great Deluge"* algorithm GD. The solution selection process helps prevent the GD from getting stuck in local optima (Rajesh et al., 2023).

In the following section, the improved cuckoo search algorithm is thoroughly explained. Later, in Section 4, the generalization capacity of the improved cuckoo search algorithm will be assessed using ten benchmark datasets, and the obtained results will be discussed in detail. Furthermore, in Section 5, the suggested algorithm will be used for building IDSs, and the performance of the produced models will be examined. The article concludes and discusses the possible future works in Section 6.

# 3 The Improved Cuckoo Search Algorithm (ICA)

The original CSA is stimulated by the cuckoo bird's force-brooding attitude (Yang & Deb, 2009). This attitude starts once the cuckoo lays its egg in the nest of another bird called a host. Cuckoo eggs normally hatch earlier than the host's eggs. After a while, the cuckoo chick discovers the presence of the host's eggs and decides to get rid of them. The three main principles in the CSA, as defined in (Dua & Graff, 2017), are:

- Every cuckoo lays a single egg in a nest randomly.
- High-quality nests will be put apart and considered for additional enhancement.
- The number of nests is known beforehand; the cuckoo chick realizes other eggs in every nest depending on a probability value ranging from 0 to 1. After that, the host decides to leave the nest or get rid of the egg.

Overall, the key pillars of the CSA are the nest, which denotes the population; the solutions, which correspond to the eggs inside a nest; and the cuckoo's egg, which represents a new solution identified through the Levy-flight algorithm, which is illustrated in Equation 1.

$$x_{i+1} = x_i + \alpha \oplus \text{Levy}(\lambda) \qquad (1)$$

In equation 1, the $x_{i+1}$ refers to the random step to the following location according to the present location, $x_i$ denotes the new solution for cuckoo $i$, $\alpha$ denotes the step size, and it usually is greater than 0, $\lambda$ denotes a constant of distribution, $\oplus$ indicates the entry-wise multiplication which is just like those found in the Particle Swarm Optimization algorithm (PSO) except that the random step in (1) explores the search space in a better way as the step size is eventually longer.

The Levy flight offers a random step, and its size is driven from the Levy distribution in Equation 2.

$$Levy{\sim}u = t^{-\lambda},\ Where\ 1 < \lambda \leq 3 \tag{2}$$

The new solution will then be compared with other solutions, and the better solution(s) will be substituted by the bad solution(s). This is tricky because rejecting a possible solution might result in a minor fraction. However, the selected solution might be found later as a non-useful solution. On the other hand, the rejected solution might be shown to be a visible solution in the future. Therefore, one of the possible improvements to the CSA is to accept bad solutions.

Nevertheless, the acceptance of bad solutions should be implemented carefully. Here, using an optimization algorithm might be justified. The CSA algorithm is presented in Figure 1. Water behavior has also inspired researchers to create attribute selection algorithms (Dueck, 1993) (Alzaqebah et al., 2021). The GD was one of the first algorithms developed in this regard. GD mimics the hill climber behavior who tries to keep his feet dry during a great deluge. GD will accept the solution(s) with bad objective values according to the water level (border value). During the search process, the border value will be reduced with a decay ratio. Decreasing the level value causes the solution(s) to decrease continuously until convergence.

**Algorithm 1.** *Cuckoo search algorithm*

```
Objective function $f(\vec{x}), \vec{x} = (x_1, x_2, \cdots, x_d)^T$
Generation $t = 1$
Initial a population of **n** host nests $x_i$ $(i = 1,2,\ldots,n)$
**While** $(t < Maximum\ Generation)$ or (stop criterion)
    Get a cuckoo (say $i$) randomly by Lévy flights
    Evaluate fitness for cuckoo $F$
    Choose a nest among $n$(say $j$) randomly
    **if** $(F_i > F_j)$ **then**
        Replace $j$ by the new solution
    **End if**
    Abandon a faction $(P_a)$ of worse nests and build new ones
    Keep the best solutions (or nests with quality solutions)
    Rank the solutions and find the current best
    Update the generation number $t = t + 1$
**End while**
```

Figure 1: Pseudo Code of CSA (Javidi et al., 2021)

**Algorithm 2.**
Great Deluge Algorithm

```
Set estimated quality of every solution, $EstimatedQuality = f(Sol_i) - F_i$,
  where ($i = 1, \ldots, population\ size$) and $F$ is a total force taken from Figure 3;
Calculate force decay rate, $\boldsymbol{\beta} = EstimatedQuality/NumOfIte\_GD$;
$Set\ iteration\_GD \leftarrow 0$;
for ($iteration\_GD < NumOfIte\_GD$)
    Define a randomly selected neighbourhood structure ($Nbs1$, or $Nbs2$) on $Sol$ to
    generate a new solution called $Sol*$;
    Calculate $f(Sol*)$;
    if ($f(Sol*) < f(Solbest)$)
        $Sol \leftarrow Sol*$;
        $Sol_{best} \leftarrow Sol^*$:
    else
        if ($f(Sol*) \leq$ level)
            $Sol \leftarrow Sol*$;
        endif
    endif
    $level = level - \boldsymbol{\beta}$;
    Increase $iteration\_GD$ by $1$;
end for
```

Figure 2: Pseudo Code of GD (Talbi, 2009)

The GD is a recursive process whereby each iteration produces k-neighbor solutions. Then it will assess all possible solutions and pick only the solutions that are better than the existing ones or if they are equal to or less than the border level. The pseudocode of GD is shown in Figure 2. In this research, the ICA will be used to select the ideal feature group among the whole attribute set in a given dataset. The solution for the attribute selection problem could be represented as an array of size N filled with either 0 or 1, in which N denotes the total number of attributes within a training dataset, 0 implies that the attribute is not selected, on the other hand, 1 means that the attribute is specified. As stated earlier, a tiny fraction may result in CSA rejecting a possible feature selection solution and initiating searching for a new solution. Solution rejection might be a time-consuming process.

However, in some cases, there might not be enough iterations left for searching for a new solution, resulting in the CSA being stuck in local optima. Therefore, an improved strategy is needed before rejecting a possible solution by accepting bad-looking neighbor solutions. This research suggests an improved solution selection strategy using the GD algorithm with 2- neighbor solutions. This will accelerate the CSA convergence and prevent it from being stuck in local optima. More information about possible neighborhood strategies can be found in (Alzaqebah et al., 2022). Let us assume a solution array as follows:

SOL= [0,1,0,1,1,0,1,1,0,0]. Hence, the proposed 2-neighbor strategy applies the following principles:

- Neighbor move strategy: randomly select a feature and move it to another random position.
- Neighbor swap strategy: randomly select two features and swap their values.

The ICA is depicted in Figure 3.

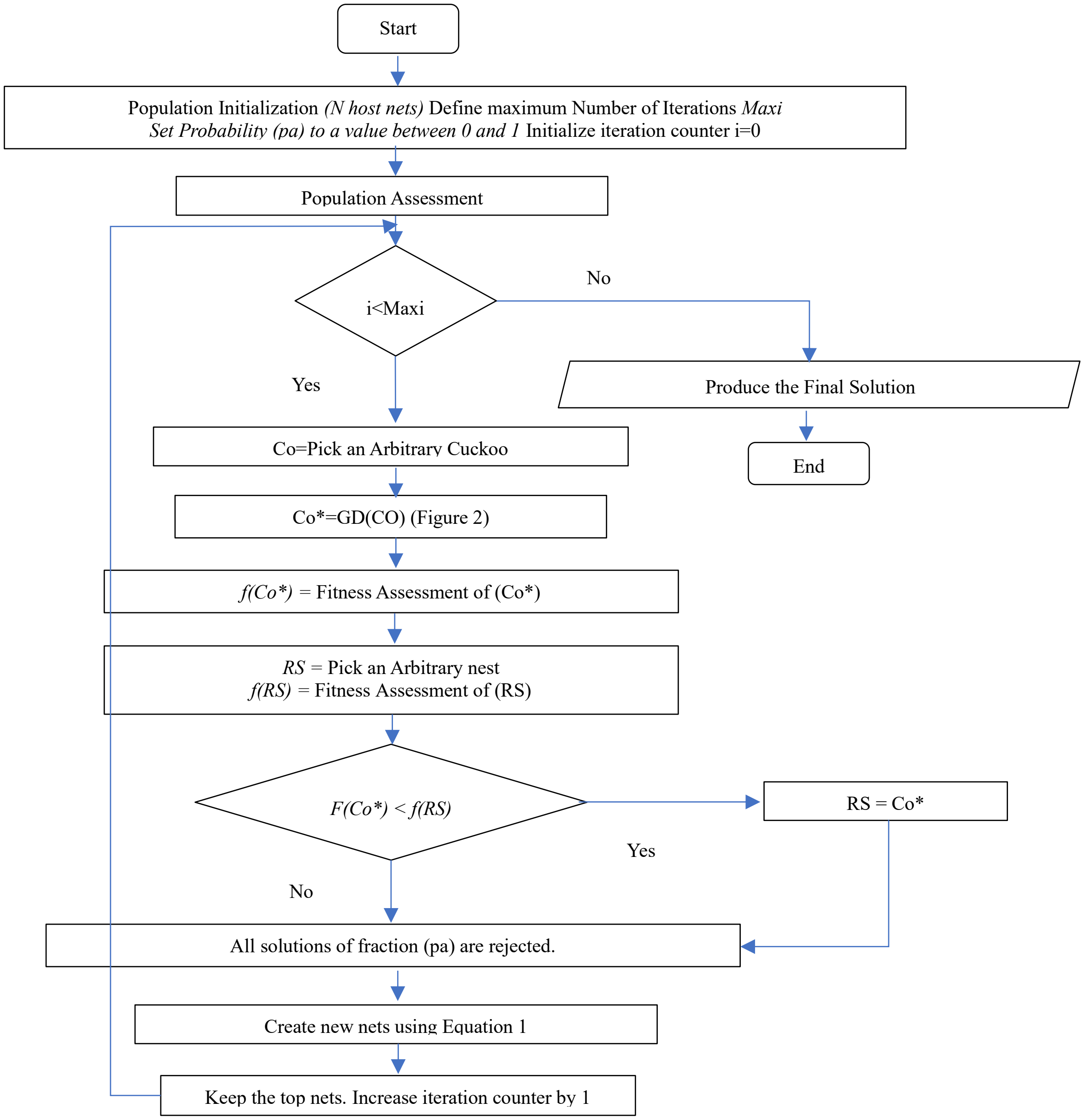
Start
Population Initialization (N host nets) Define maximum Number of Iterations Maxi
Set Probability (pa) to a value between 0 and 1 Initialize iteration counter i=0
Population Assessment
i<Maxi
No
Yes
Produce the Final Solution
End
Co=Pick an Arbitrary Cuckoo
Co*=GD(CO) (Figure 2)
f(Co*) = Fitness Assessment of (Co*)
RS = Pick an Arbitrary nest
f(RS) = Fitness Assessment of (RS)
F(Co*) < f(RS)
RS = Co*
Yes
No
All solutions of fraction (pa) are rejected.
Create new nets using Equation 1
Keep the top nets. Increase iteration counter by 1

Figure 3: The Improved Cuckoo Search Algorithm

The uniqueness of this algorithm is that it employs the GD algorithm for selecting the possible solutions, and it accepts bad steps based on a probability fraction *(pa)*. This means it tolerates bad solutions with a specific probability of investing the time in the best manner and avoiding the CSA of getting stuck in local optima.

## 4 Generalization Ability of the ICA Using Benchmark Datasets

The ICA is implemented and embedded in WEKA (Hall et al., 2011). All experiments were executed on a PC with MS Windows 10, 16 gigabytes of RAM, and an i7 processor @ 1.90 GHz. A couple of experiments will be carried out. This section assesses the ICA's generalization capacity using ten benchmark UCI datasets (Dua & Graff, 2017). To evaluate the ability of ICA under diverse conditions, the chosen benchmark datasets have different examples, attributes, and values in the class variable, as indicated in Table 1.

Table 1: Description of Benchmark Datasets

| Dataset Name | #of Examples | #of attributes | #of Classes Values |
|---|---|---|---|
| Hepatitis | 155 | 20 | 2 |
| Segment | 2310 | 20 | 7 |
| German Credit Credit-g” | 1000 | 21 | 2 |
| Mushroom | 8124 | 23 | 2 |
| Autos | 205 | 26 | 7 |
| Anneal | 898 | 39 | 6 |
| Audiology | 226 | 70 | 24 |
| Wine | 178 | 14 | 3 |
| Soybean | 683 | 36 | 19 |
| Lung Cancer | 32 | 57 | 3 |

In the next section (Section 5), the ICA will be utilized for selecting the crucial attributes for constructing an intelligent model for detecting intrusion attacks. Thanks to WEKA for providing the *“Attribute-Selected-Classifier”* (ASC) as a meta-learning strategy. ASC combines two stages: the dimensionality reduction phase and the classifier creation phase. Firstly, the ASC selects the ideal features using a specific selection technique. The selected attributes will be then used in the second phase for building a classification model. Although ICA can be used in filter-based and wrapper-based attribute selection strategies, the filter-based approach in this research will be utilized.

Nevertheless, using a filter-based approach requires identifying the attribute evaluator strategy. The *“cfsSubsetEval”* has been utilized in this research as a feature evaluator. This strategy assesses the importance of a subset of features by looking at every attribute’s predictive capability and the redundancy degree among the attributes within the subset. Several pilot tests have been completed to realize the ICA’s optimal parameter settings. Table 2 shows the adopted parameter settings that were attained from the pilot tests and were used for building classification models using the ICA.

Table 2: Employed Parameters

| Parameter Name | Value |
|---|---|
| Pa | 0.25 |
| $\lambda$ | 2 |
| Population Size | 20 |
| Iterations | 100 |

On the other hand, the default parameters were used when using different considered algorithms. The Naïve Bayes algorithm is employed in all experiments because it applies simple calculations based on the *“Bayes Theorem”*. The outcomes obtained from ICA were contrasted with several nature-inspired attribute selection algorithms, including *“ANT Colony”* (Aghdam et al., 2009), *“BAT”* (Nakamura et al., 2012), and “*Bee Colony”* (Rao et al., 2019).

Table 3: Empirical Results Obtained from ICA using Benchmark Datasets (Acc=Accuracy, Pre=Presession, Rec=Recall, F1=F1-Score)

| | **ICA** | | | | **ANT** | | | | **BAT** | | | | **Bee** | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | **Acc** | **Pre** | **Rec** | **F1** | **Acc** | **Pre** | **Rec** | **F1** | **Acc** | **Pre** | **Rec** | **F1** | **Acc** | **Pre** | **Rec** | **F1** |
| Hepatitis | **86.5%** | **87.0%** | **86.5%** | **86.7%** | 83.9% | 84.9% | 83.9% | 84.3% | 82.6% | 83.7% | 82.6% | 83.0% | 84.5% | 84.9% | 84.5% | 84.7% |
| Segment | 87.5% | 87.7% | 87.5% | 87.6% | **88.2%** | **88.5%** | **88.2%** | **88.2%** | 86.1% | 86.1% | 86.1% | 86.0% | 86.2% | 86.2% | 86.2% | 86.2% |
| Credit-g | **74.4%** | **72.7%** | **74.4%** | **72.5%** | 73.8% | 71.9% | 73.8% | 71.8% | 73.4% | 71.5% | 73.4% | 71.4% | **74.4%** | **72.7%** | **74.4%** | **72.5%** |
| Mushroom | **98.9%** | **98.9%** | **98.9%** | **98.9%** | **98.9%** | **98.9%** | **98.9%** | **98.9%** | 98.7% | 98.7% | 98.7% | 98.7% | 98.5% | 98.6% | 98.5% | 98.5% |
| Autos | **62.0%** | **59.5%** | **62.0%** | **59.5%** | 58.0% | 55.7% | 58.0% | 55.3% | 56.6% | 53.6% | 56.6% | 54.5% | 55.1% | 55.2% | 55.1% | 53.1% |
| Anneal | **87.3%** | **93.3%** | **87.3%** | **89.0%** | 86.5% | 92.8% | 86.5% | 88.2% | 85.9% | 92.6% | 85.9% | 87.8% | 86.8% | 91.5% | 86.9% | 88.0% |
| Audiology | **73.0%** | **70.4%** | **86.4%** | **77.6%** | 70.8% | 59.3% | 72.7% | 65.3% | 70.8% | 58.3% | 63.6% | 60.9% | 72.1% | 60.7% | 77.3% | 68.0% |
| Wine | **97.8%** | **97.8%** | **97.8%** | **97.7%** | 97.2% | 97.3% | 97.2% | 97.2% | 97.2% | 97.3% | 97.2% | 97.2% | 97.2% | 97.3% | 97.2% | 97.2% |
| Soybean | 92.4% | 93.1% | 92.4% | 92.3% | 92.1% | 93.1% | 92.1% | 92.1% | **92.7%** | **93.6%** | **92.7%** | **92.6%** | 92.1% | 93.1% | 92.1% | 92.0% |
| Lung Cancer | **68.8%** | 70.5% | **68.8%** | **69.2%** | **68.8%** | **70.8%** | **68.8%** | 68.9% | 56.3% | 58.0% | 56.3% | 56.8% | 62.5% | 65.3% | 62.5% | 63.0% |

The selection of these algorithms is because they showed promising results when applied to several domains (Rais & Mehmood, 2018; Zitouni et al., 2020; Contreras-Cruz et al., 2015). Four evaluation criteria were adopted to compare the overall performance of ICA and other considered attribute selection algorithms: Accuracy, Precession, Recall, and F1-Score. Further information about how these metrics can be calculated is be found in (Mohammad et al., 2014). The validation technique employed in all experiments was the 10-fold cross-validation approach (Mohammad et al., 2014). The empirical outcomes are illustrated in Table 3. Digging deep into the results, it has been shown that the ICA could notably reduce the number of attributes in all cases, as depicted in Table 4. The ICA picked fewer features than ANT, BAT, and Bee in 5, 8, and 2 cases, respectively. These results substantiate the usefulness of the suggested attribute selection strategy offered in ICA, which considers accepting bad steps based on probability fraction *(pa)* towards selecting the most influential group of attributes. However, the ICA determined more attributes than ANT, BAT, and Bee in 1, 1, and 5 cases, respectively. However, although the ICA algorithm selects more features, it produces comparable performance outcomes in some cases. This confirms the selected features' effectiveness due to the effective strategy employed in ICA. In other words, merging CSA with GD is a good approach for selecting the most beneficial attributes for building intelligent models. A Win-Tie-Loss count table (Table 5) is created to provide a thorough cross-classifier comparison.

Table 4: Number and Ratio of Selected Features (A*=Number of Selected Features, Ration=Features Reduction Rate)

| | **ICA** | | **ANT** | | **BAT** | | **Bee** | |
|---|---|---|---|---|---|---|---|---|
| | **A*** | **Ratio** | **A*** | **Ratio** | **A*** | **Ratio** | **A*** | **Ratio** |
| Hepatitis | 10 | 47.4% | 10 | 47.4% | 10 | 47.4% | 10 | 47.4% |
| Segment | 7 | 63.2% | 9 | 52.6% | 8 | 57.9 | 7 | 63.2% |
| Credit-g | 4 | 80.0% | 3 | 85.0% | 3 | 85.0% | 3 | 85.0% |
| Mushroom | 5 | 77.3% | 5 | 77.3% | 6 | 72.7% | 4 | 81.8% |
| Autos | 6 | 76.0% | 6 | 76.0% | 8 | 68.0% | 5 | 80.0% |
| Anneal | 10 | 73.7% | 10 | 73.7% | 12 | 68.4% | 10 | 73.7% |
| Audiology | 17 | 75.4% | 23 | 66.7% | 23 | 66.7% | 16 | 76.8% |
| Wine | 10 | 23.1% | 11 | 15.4% | 11 | 15.4% | 11 | 15.4% |
| Soybean | 22 | 38.9% | 25 | 30.6% | 27 | 25.0% | 24 | 33.3% |
| Lung Cancer | 12 | 78.6% | 15 | 73.2% | 17 | 69.6% | 9 | 83.9% |

The results in Table 5 showed that the ICA algorithm uniquely produced higher Accuracy, Precision, Recall, and F1-Score than ANT, BAT, and Bee in 5, 5, 5, and 6 cases respectively.

Nonetheless, the ICA algorithm produced the highest Accuracy, Precision, Recall, and F1-Score in 8, 7, 8, and 8 instances, whereas it had the same results as some other algorithms in 3, 2, 3, and 2 cases respectively. Nevertheless, the ICA produced less Accuracy, Precision, Recall, and F1-Score than ANT, BAT, and Bee in 2, 3, 2, and 2 cases respectively.

All attribute selection approaches aim to enhance or maintain the models' performance compared to those created using the entire dataset. In other words, any attribute selection strategy must produce compact models and improve or preserve the produced model(s) performance. Given that, Figure 4 compares the models created using ICA and the ones made using the entire dataset. In this figure, a positive ratio means that the ICA produced a better result.

Table 5: Win-Tie-Loss Count Table (W=Win, T=Tie, L=Loss)

| | Acc | | | Pre | | | Rec | | | F1 | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | **W** | **T** | **L** | **W** | **W** | **L** | **W** | **W** | **L** | **W** | **W** | **L** |
| ICA | **5** | 3 | 2 | **5** | 3 | 3 | **5** | 3 | 2 | **6** | 2 | 2 |
| ANT | 1 | 5 | 7 | 2 | 4 | 7 | 1 | 4 | 7 | 1 | 2 | 8 |
| BAT | 1 | 2 | 9 | 1 | 1 | 9 | 1 | 1 | 9 | 1 | 1 | 9 |
| Bee | 0 | 3 | 9 | 0 | 4 | 9 | 0 | 3 | 9 | 0 | 2 | 9 |

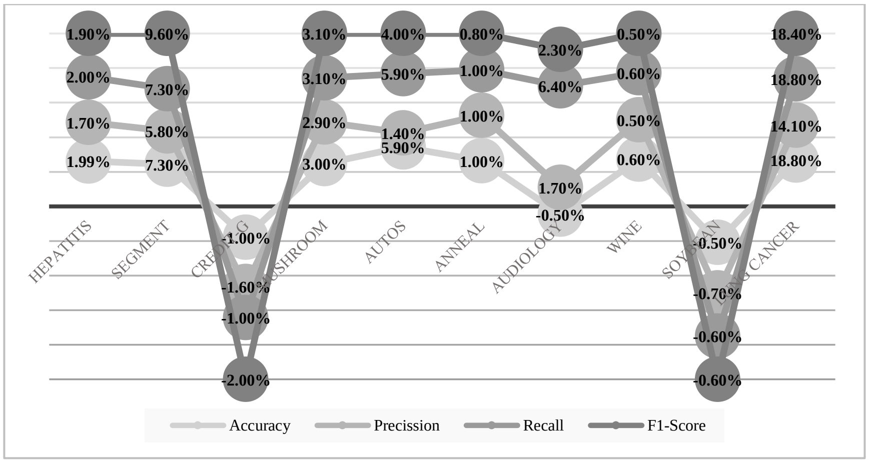

Figure 4: Performance Comparison between Models Created using ICA and Full Dataset

Figure 4 confirmed that ICA could attain the trade-off between producing compact and high-accuracy models simultaneously. It is vital to note that the more compacted the model is, the faster the model creation will be and the faster the model is in doing the required calculations to produce the final result (Class assigning). Notably, ICA produced models with less overall performance when applied to *"Credit-g"* and *"Soyabeans"* datasets. Surprisingly, all other attribute selection algorithms tend to have the same results as the ICA algorithm when applied to these datasets since they produced less accurate models than the models produced using the complete dataset. A justification could be that in addition to being imbalanced, these datasets suffer from misclassification errors on the minority class, i.e., the class value that appears less.

However, the results obtained from ICA are still acceptable. To elaborate, the ICA produced less Accuracy, Precision, Recall, and F1-Score than when using the entire dataset by 1%, 1.6%, 1%, 2%, and 0.5%, 0.7%, 0.6%, 0.6% on Credit-g and Soyabeans datasets respectively. These fractions could be essential in some domains, and the model designer might sacrifice the compactness to produce more accurate models. However, in other domains, such as cybersecurity, the model designer might prefer to create more compact models to assign the class value as quickly as possible because the slower the decision-making is, the more likely the attack might succeed. Hence, the less confident end users will be in the intelligent models in providing a secure environment. This indeed stimulated doing some experiments to explore how effective the ICA is in producing IDSs. This will be the primary goal of the experiments that will be done in the next section.

# 5 Intrusion Detection using ICA

The *"NSL-KDD"* dataset (Saporito, 2019) is employed in this section for building intelligent IDS models. Even though it consists of an identical group of attributes like in *"KDD 99" (Stolfo et al., 2000)*, the *"NSL-KDD"* is a better as well as a minimized version. Also, it tackles some problems acquired from the *"KDD 99"*. A worthwhile change is that it does not include repetitive examples; consequently, the developed model(s) may not be inclined toward the more recurring examples. This improvement is vital throughout the attribute selection procedure since repetitive examples will probably lead to picking several ineffective attributes that were chosen simply because they appeared more frequently.

The *"NSL-KDD"* is comprised of forty-one attributes and another class attribute. Detailed information on such attributes is available in (Saporito, 2019). The class feature includes twenty-one possible values, which are categorized as four kinds of attacks those are:

- Denial of Service attacks (*"DoSs"*): meant to switch off service and make it unavailable.
- User to Root attacks (*"U2R"*): an attempt to get into users' accounts to get access to root details.
- Remote to Local attack (*"R2L"*): attempts to retrieve a particular network address illegitimately remotely.
- Prob-Response attacks (Probing): deliberately crafted, so the victims discover and report it. Attackers may then find the locations of the detection and security systems that use such reports.

A couple of versions were offered of the *"NSL-KDD"*, a multiple-class dataset, and a two-class dataset. The class attribute in the multiple class dataset has a total of five values, namely *"DoS", "U2R", "R2L", "Probe"*, and *"Normal"* which are shown 45927, 52, 995, 11656, and 67343, correspondingly. Whereas the two-class dataset involves two class values where 67343 of the dataset examples belong to *"Normal"* and the remaining 58630 are *"Anomaly"*. The two-class dataset is well-balanced. On the other hand, the multiple-class dataset is an unbalanced one.

The ICA will be evaluated on both datasets, and the outcomes will be discussed thoroughly. Commonly, IDSs apply several rules to detect vulnerabilities. Different strategies for producing these rules are generally employed, including human knowledge and trial and error strategies. Intelligent rule-based methods might be utilized to set up a group of rules that could be employed in constructing IDS systems. So, besides using *"Naïve Bayes"*, a popular rule-based intelligent algorithm will be employed in this section, that is, the Decision Tree (*"DT"*). The same parameter settings depicted in Table 2 (Section 4) will be employed for building the model. It is worth noting that the set of experiments completed in this section will not use the ASC as in Section 4. In other words, the experiments start by selecting a set of significant features, and the resulting datasets are then used to create intelligent IDS models. The rationale behind this practice is to compare the time needed for creating the IDS models before and after the attribute selection phase. The time required for producing the models is assessed in this section because the faster the model creation is, the more reliable the intelligent cybersecurity models are in the cybersecurity domain.

### Attained Results from the Two Class Dataset

The experimental results showed that ICA selected nine attributes; those features are *"flag", "src_bytes", "dst_bytes", "logged_in", "srv_serror_rate", "same_srv_rate", "diff_srv_rate", "dst_host_srv_diff_host_rate", and "dst_host_srv_serror_rate"*. This represents a reduction in the number of features by almost 78.00%. Despite the significant attribute reduction rate, the models created using the *"Naïve Bayes",* and the *"DT"* should preserve or maintain the performance of the models developed using the entire dataset. Therefore, the performance of the models before and after the attribute selection process are evaluated. Considering the results obtained from the *"DT"* (Figure 5), it has been shown that the performance of the model developed using the features generated by ICA has reduced by 0.18%, 0.20%, 0.20%, and 0.20% with respect to Accuracy, Precision, Recall, and F1-Score respectively.

The difference is minor in all cases, and it can be tolerated under some circumstances, i.e., if the produced model is more compact or/and if the time needed for generating the model is shorter. Therefore, the time required for constructing the models and the number of rules generated before and after applying the ICA were investigated. The results showed that the time needed to build the models after using the ICA has significantly reduced from 27.53 seconds to 3.9 seconds. This constitutes a reduction of 85.83%, which means that the ICA can decrease the time required for creating intelligent models, which is very important in the field of cybersecurity in general and in IDSs in particular, considering that there is a constant race between the attackers and security systems since the set of significant attributes are constantly changing over time which requires building new models occasionally.

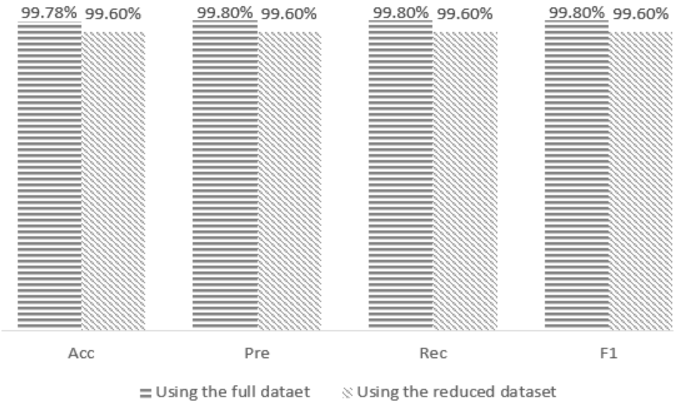

Figure 5: Performance of DT Using the Entire Dataset and the Reduced Dataset

Furthermore, it has been shown that the number of rules was reduced from 605 rules to 149 rules, constituting a reduced rate of almost 75%. This will accelerate the decision-making process (class prediction) because fewer rules will be explored during the class prediction phase. This is particularly important because every second is important to prevent intrusion attempts and protect the end users. In general, the models produced using the dataset generated by the ICA proved efficient. On the other hand, the results obtained from *"Naïve Bayes"* before and after applying the ICA are depicted in Figure 6.

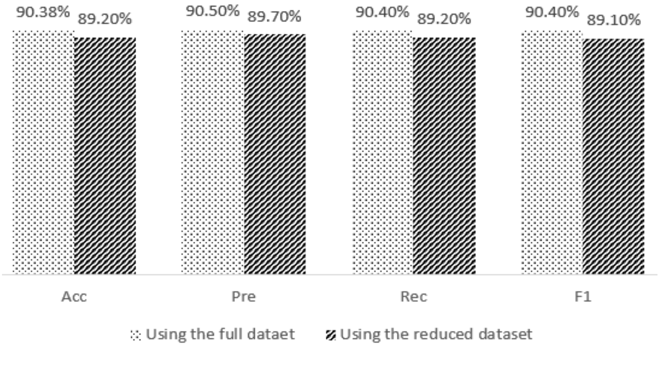

Figure 6: Performance of NB Using the Entire Dataset and the Reduced Dataset

In this figure, it has been shown that the *"DT"* produced improved results than *"Naïve Bayes"* in all situations. For instance, the *"Naïve Bayes"* made less Accuracy, Precision, Recall, and F1-Score by ratios of 9.40%, 9.30%, 9.40%, and 9.40%, respectively, when using the entire dataset and with percentages of 10.40%, 9.90%, 10.40%, and 10.50% respectively when using the dataset produced by the ICA. However, the *"Naïve Bayes"* generated better outcomes when employing the full dataset with respect to Accuracy, Precision, Recall, and F1-Score with ratios of 1.18%, 0.80%, 1.20%, and 1.30%, respectively. Nevertheless, the time needed for producing the *"Naïve Bayes"* models has reduced from 1.07 seconds when using the entire dataset to 0.19 seconds when using the dataset produced from the ICA.

**Attained Results from the Multi-Class Dataset**

The results attained from the multiclass dataset were very interesting. For instance, the number of selected attributes using the ICA was 11 attributes those are *"service", "flag", "src_bytes", "dst_bytes", "logged_in", "root_shell", "srv_serror_rate", "same_srv_rate", "diff_srv_rate", "dst_host_srv_diff_host_rate", and "dst_host_serror_rate"*. This means two extra features are shown to be significant compared to the two-class values dataset: "*service",* and *"root_shell"*. Regarding the overall performance of the constructed model(s), the outcomes revealed that the models developed using the entire dataset slightly surpass the model built using the dataset produced from the ICA with respect to Accuracy, Precision, Recall, and F1-Score by ratios of 0.03%, 0.10%, 0.10%, and 0.10% respectively when using the *"DT"* and with ratios of 0.06%, 2.50%, 0.10%, and 2.30% respectively when using the *"Naïve Bayes"*. Nevertheless, the time required for constructing the models has significantly dropped from 33.85 seconds to 5.81 seconds when using the *"DT"* and from 0.97 seconds to 0.27 seconds when using the *"Naïve Bayes"*. Surprisingly, the *"DT"* model made from the dataset produced by the ICA provided 472 extra rules than the model constructed using the entire dataset.

However, this might be a good sign that the model discovered helpful patterns in the data regardless of the reduced attribute set. An unbalanced dataset might emerge as a major cause of generating additional rules because the model tries to use the group of selected attributes to cover all possible class values, which might be achieved only by developing more rules. The lowest accuracy rate was achieved in predicting the *"R2L"* attack. However, the number of examples associated with this type of attack was 52 instances only when using the *"DT", w*hich means that the model does not have enough samples to learn. Yet, the highest accuracy by the *"DT"* is achieved for the *"Normal"* class considering that this class has the highest number of instances. Therefore, it has been clearly shown that class balancing is an important step toward generating intelligent models using the multiclass dataset.

The empirical analysis showed that the ICA could pick a set of attributes for generating IDSs. This motivates exploring the applicability of ICA in other cybersecurity domains in the near future.

## 6 Conclusions and Future Work

There are several factors affecting the quality of any intelligent IDS. One of these factors is the time required for constructing the IDSs. To elaborate, it is well-known that the attributes that distinguish normal attempts from suspicious ones constantly change over time. This is because the attackers know that the longer they use the same technique and the same attributes, the more likely the security experts will be able to create systems for detecting such attacks. Therefore, it is crucial to cope with any possible change in the attribute set by responding quickly to these changes by constructing new IDSs. The time required for building (training) a new model depends heavily on the dimensionality of the training dataset, i.e., the number of instances and attributes. Therefore, the smaller the number of attributes, the shorter the time required to train a new model. Another factor affecting IDS quality of IDSs is the time needed to decide if an attempt is normal or an attack. Again, this factor is greatly affected by the model complexity, which is directly affected by the number of attributes used in the decision-making process.

Consequently, the fewer attributes, the less complex the model is. Furthermore, the accuracy of the IDS plays a vital role in producing high-quality IDSs. Yet, the more attributes employed for building IDSs do not necessarily mean the more accurate the model will be. In some cases, the more attribute set might mislead the intelligent IDS and result in low-quality models due to overfitting. All these reasons (and others) motivated exploring the importance of attribute selection algorithms during the preprocessing stage while building new intelligent IDSs. Several attribute selection algorithms were proposed in the literature. However, what makes a feature selection algorithm different is the selection of high-quality attributes that would help produce better IDSs. This research bundled two nature-inspired attribute selection algorithms: the cuckoo algorithm and the great deluge algorithm. Such an algorithm is named an Improved Cuckoo Search Algorithm (ICA), later used for building an Intrusion Detection model using an Improved Cuckoo Search Optimization Algorithm. The uniqueness of the ICA is that it tolerates some bad steps with a specific probability towards selecting the most critical set of attributes. The generalization capacity of the ICA has been assessed using ten benchmarking datasets, and the results proved the quality of the attributes selected by the ICA. This was clearly shown because the classification models created using the ICA produced better results in eight out of ten cases. Yet, the other two cases have some particularity in that these datasets suffer from misclassification errors in the minority class as a direct result of imbalanced datasets. Later, the ICA is used for constructing intelligent IDS using the *"NSL-KDD"* datasets. Two experiments were accomplished because such a dataset has two versions, i.e., two-class and multiclass datasets. The *"Decision Tree Algorithm"* and the *"Naïve Bayes Algorithm"* were employed to construct the IDS systems after obtaining the reduced dataset from the ICA. The results were very appealing regarding accuracy, precession, recall, and F1-Score, the time required to construct the IDS systems, and the number of generated rules.

Nevertheless, the intelligent models constructed using the two-class dataset gave better outcomes. This is because this dataset is well-balanced compared to the multiclass dataset. Consequently, dataset balancing plays a significant role in producing better IDSs. As a result, a possible future research direction is to incorporate dataset dimensionality reduction techniques with feature selection algorithms. Moreover, applying the ICA algorithm to other cybersecurity domains is a potential research direction to assess the performance of the ICA under different situations.

### Funding

This research was funded by the SAUDI ARAMCO Cybersecurity Chair at Imam Abdulrahman Bin Faisal University (IAU).

### Availability of Data and Materials

The dataset is publicly available at: https://www.unb.ca/cic/datasets/nsl.html

## References

[1] Abdullah, M., Alshannaq, A., Balamash, A., & Almabdy, S. (2018). Enhanced intrusion detection system using feature selection method and ensemble learning algorithms. *International Journal of Computer Science and Information Security (IJCSIS), 16*(2), 48-55.

[2] Abualigah, L. M., Khader, A. T., & Hanandeh, E. S. (2018). A new feature selection method to improve the document clustering using particle swarm optimization algorithm. *Journal of Computational Science, 25,* 456-466.

[3] Aghdam, M. H., Ghasem-Aghaee, N., & Basiri, M. E. (2009). Text feature selection using ant colony optimization. *Expert systems with applications, 36*(3), 6843-6853.

[4] Almasoudy, F. H., Al-Yaseen, W. L., & Idrees, A. K. (2020). Differential evolution wrapper feature selection for intrusion detection system. *Procedia Computer Science, 167,* 1230-1239.

[5] Alzaqebah, M., Briki, K., Alrefai, N., Brini, S., Jawarneh, S., Alsmadi, M. K., & Alqahtani, A. (2021). Memory based cuckoo search algorithm for feature selection of gene expression dataset. *Informatics in Medicine Unlocked, 24,* 100572. https://doi.org/10.1016/j.imu.2021.100572

[6] Alzaqebah, M., Jawarneh, S., Alwohaibi, M., Alsmadi, M. K., Almarashdeh, I., & Mohammad, R. M. A. (2022). Hybrid brain storm optimization algorithm and late acceptance hill climbing to solve the flexible job-shop scheduling problem. *Journal of King Saud University-Computer and Information Sciences, 34*(6), 2926-2937.

[7] Ambusaidi, M. A., He, X., Tan, Z., Nanda, P., Lu, L. F., & Nagar, U. T. (2014). A novel feature selection approach for intrusion detection data classification. *In IEEE 13th international conference on trust, security and privacy in computing and communications,* 82-89.

[8] Asl, T. M., & Asl, T. S. (2022). Strategy Optimization for Responding to Primary, Secondary and Residual Risks Considering Cost and Time Dimensions in Petrochemical Projects. *Archives for Technical Sciences, 2*(27), 33-48.

[9] Aziz, M. A. E., & Hassanien, A. E. (2018). Modified cuckoo search algorithm with rough sets for feature selection. *Neural Computing and Applications, 29,* 925-934.

[10] Babenko, V., Danilov, A., Vasenin, D., & Krysanov, V. (2021). Parametric Optimization of the Structure of Controlled High-voltage Capacitor Batteries. *Archives for Technical Sciences, 1*(24), 9–16.

[11] Balakrishnan, S., Venkatalakshmi, K., & Kannan, A. (2014). Intrusion detection system using feature selection and classification technique. *International journal of computer science and application, 3*(4), 145-151.

[12] Bhaskar, T., Hiwarkar, T., & Ramanjaneyulu, K. (2019). Adaptive jaya optimization technique for feature selection in NSL-KDD data set of intrusion detection system. *In Proceedings of International Conference on Communication and Information Processing (ICCIP).*

[13] Bhumgara, A., & Pitale, A. (2019). Detection of network intrusions using hybrid intelligent systems. *In IEEE 1st International Conference on Advances in Information Technology (ICAIT),* 500-506.

[14] Camgözlü, Y., & Kutlu, Y. (2023). Leaf Image Classification Based on Pre-trained Convolutional Neural Network Models. *Natural and Engineering Sciences, 8*(3), 214-232.

[15] Contreras-Cruz, M. A., Ayala-Ramirez, V., & Hernandez-Belmonte, U. H. (2015). Mobile robot path planning using artificial bee colony and evolutionary programming. *Applied Soft Computing, 30,* 319-328.

[16] Cover, T. M. (1999). *Elements of information theory.* John Wiley & Sons.

[17] Dua, D., & Graff, C. (2017). UCI machine learning repository. *University of California, Irvine, School of Information and Computer Sciences.* http://archive.ics.uci.edu/m

[18] Dueck, G. (1993). New optimization heuristics: The great deluge algorithm and the record-to-record travel. *Journal of Computational physics, 104*(1), 86-92.

[19] Elshrkawey, M., Alalfi, M., & Al-Mahdi, H. (2021). An enhanced intrusion detection system based on multi-layer feature reduction for probe and dos attacks. *Journal of Internet Services and Information Security , 11*(4), 61-78.

[20] Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., & Witten, I. H. (2011). Waikato environment for knowledge analysis. *University of Waikato.* http://www.cs.waikato.ac.nz/ml/weka/

[21] Javidi, Z., Akbari, R., & Bushehrian, O. (2021). A new method based on formal concept analysis and metaheuristics to solve class responsibility assignment problem. *Iran Journal of Computer Science, 4*(4), 221-240.

[22] Kannan, A., Venkatesan, K. G., Stagkopoulou, A., Li, S., Krishnan, S., & Rahman, A. (2015). A novel cloud intrusion detection system using feature selection and classification. *International Journal of Intelligent Information Technologies (IJIIT), 11*(4), 1-15.

[23] Karim, M. R., & Kaysar, M. M. (2016). *Large scale machine learning with spark.* Packt Publishing.

[24] Kumar, K., & Batth, J. S. (2016). Network intrusion detection with feature selection techniques using machine-learning algorithms. *International Journal of Computer Applications, 150*(12).

[25] Lin, K. C., Zhang, K. Y., Huang, Y. H., Hung, J. C., & Yen, N. (2016). Feature selection based on an improved cat swarm optimization algorithm for big data classification. *The Journal of Supercomputing, 72,* 3210-3221.

[26] Liu, H., Zhou, M., & Liu, Q. (2019). An embedded feature selection method for imbalanced data classification. *IEEE/CAA Journal of Automatica Sinica, 6*(3), 703-715.

[27] Mkuzangwe, N. N. P., & Nelwamondo, F. V. (2017). A fuzzy logic based network intrusion detection system for predicting the TCP SYN flooding attack. *In Intelligent Information and Database Systems: 9th Asian Conference, ACIIDS 2017, Kanazawa, Japan, Proceedings, Part II 9,* 14-22. Springer International Publishing.

[28] Mohammad, R. M. (2018). A neural network based digital forensics classification. *In IEEE/ACS 15th International Conference on Computer Systems and Applications (AICCSA),* 1-7.

[29] Mohammad, R. M. A. (2020). A lifelong spam emails classification model. *Applied Computing and Informatics, (ahead-of-print),* 35-54.

[30] Mohammad, R. M. A. (2020). An Improved Multi-Class Classification Algorithm based on Association Classification Approach and its Application to Spam Emails. *IAENG International Journal of Computer Science, 47*(2).

[31] Mohammad, R. M. A. (2022). An enhanced multiclass support vector machine model and its application to classifying file systems affected by a digital crime. *Journal of King Saud University-Computer and Information Sciences, 34*(2), 179-190.

[32] Mohammad, R. M. A., & Alqahtani, M. (2019). A comparison of machine learning techniques for file system forensics analysis. *Journal of Information Security and Applications, 46,* 53-61.

[33] Mohammad, R. M. A., & Alsmadi, M. K. (2021). Intrusion detection using Highest Wins feature selection algorithm. *Neural Computing and Applications, 33*(16), 9805-9816.

[34] Mohammad, R. M., Thabtah, F., & McCluskey, L. (2014). Intelligent rule-based phishing websites classification. *IET Information Security, 8*(3), 153-160.

[35] Mohammad, R. M., Thabtah, F., & McCluskey, L. (2014). Predicting phishing websites based on self-structuring neural network. *Neural Computing and Applications, 25,* 443-458.

[36] Mohammed, M. N., & Sulaiman, N. (2012). Intrusion detection system based on SVM for WLAN. *Procedia Technology, 1,* 313-317.

[37] Nakamura, R. Y., Pereira, L. A., Costa, K. A., Rodrigues, D., Papa, J. P., & Yang, X. S. (2012). BBA: a binary bat algorithm for feature selection. *In IEEE 25th SIBGRAPI conference on graphics, patterns and images,* 291-297.

[38] Nikitina, V., Raúl, A. S., Miguel, A. T. R., Walter, A. C., Anibal, M. B., Maria, D. R. H., & Jacqueline, C. P. (2023). Enhancing Security in Mobile Ad Hoc Networks: Enhanced Particle Swarm Optimization-driven Intrusion Detection and Secure Routing Algorithm. *Journal of Wreless Mobile Networks, Ubiquitous Computing, and Dependable Applications, 14*(3), 77-88.

[39] Othman, S. M., Ba-Alwi, F. M., Alsohybe, N. T., & Al-Hashida, A. Y. (2018). Intrusion detection model using machine learning algorithm on Big Data environment. *Journal of big data, 5*(1), 1-12.

[40] Pandey, S. K. (2019). Design and performance analysis of various feature selection methods for anomaly-based techniques in intrusion detection system. *Security and Privacy, 2*(1), e56. https://doi.org/10.1002/spy2.56

[41] Prasad Babu, P., & Vasumathi, A. (2023). Role of Artificial Intelligence in Project Efficiency Mediating with Perceived Organizational Support in the Indian IT Sector. *Indian Journal of Information Sources and Services, 13*(2), 39–45.

[42] Rais, H. M., & Mehmood, T. (2018). Dynamic Ant Colony System with Three Level Update Feature Selection for Intrusion Detection. *International journal of network security, 20*(1), 184-192.

[43] Rajesh, D., Giji Kiruba, D., & Ramesh, D. (2023). Energy Proficient Secure Clustered Protocol in Mobile Wireless Sensor Network Utilizing Blue Brain Technology. *Indian Journal of Information Sources and Services, 13*(2), 30–38.

[44] Rao, H., Shi, X., Rodrigue, A. K., Feng, J., Xia, Y., Elhoseny, M., & Gu, L. (2019). Feature selection based on artificial bee colony and gradient boosting decision tree. *Applied Soft Computing, 74*, 634-642.

[45] Ren, J., Guo, J., Qian, W., Yuan, H., Hao, X., & Jingjing, H. (2019). Building an effective intrusion detection system by using hybrid data optimization based on machine learning algorithms. *Security and communication networks, 2019.*

[46] Rosset, S., & Inger, A. (2000). KDD-cup 99: knowledge discovery in a charitable organization's donor database. *ACM SIGKDD Explorations Newsletter, 1*(2), 85-90.

[47] Saadawi, E. M., Abohamama, A. S., & Alrahmawy, M. F. (2024). IoT-based Optimal Energy Management in Smart Homes using Harmony Search Optimization Technique. *International Journal of Communication and Computer Technologies (IJCCTS), 12*(1), 1-20.

[48] Salim, Q. M., and Mohammed, A. E. H. (2023). Reducing False Negative Intrusions Rates of Ensemble Machine Learning Model based on Imbalanced Multiclass Datasets. *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications, 14*(2), 12-30.

[49] Saporito, G. (2019). *NSL-KDD Features*. https://docs.google.com/spreadsheets/d/1oAx320Vo9Z6HrBrL6BcfLH6sh2zIk9EKCv2OlaMGmwY/edit#gid=0

[50] Senthilnayaki, B., Venkatalakshmi, K., & Kannan, A. (2019). Intrusion detection system using fuzzy rough set feature selection and modified KNN classifier. *International Arab Journal of Information Technology, 16*(4), 746-753.

[51] Shah, R. A., Qian, Y., Kumar, D., Ali, M., & Alvi, M. B. (2017). Network intrusion detection through discriminative feature selection by using sparse logistic regression. *Future Internet, 9*(4), 81. https://doi.org/10.3390/fi9040081

[52] Shen, C., & Zhang, K. (2022). Two-stage improved Grey Wolf optimization algorithm for feature selection on high-dimensional classification. *Complex & Intelligent Systems, 8*(4), 2769-2789.

[53] Shenfield, A., Day, D., & Ayesh, A. (2018). Intelligent intrusion detection systems using artificial neural networks. *Ict Express, 4*(2), 95-99.

[54] Sreenivasu, M., Kumar, U. V., & Dhulipudi, R. (2022). Design and Development of Intrusion Detection System for Wireless Sensor Network. *Journal of VLSI Circuits and Systems, 4*(2), 1-4.

[55] Srinivasareddy, S., Narayana, Y. V., & Krishna, D. (2021). Sector beam synthesis in linear antenna arrays using social group optimization algorithm. *National Journal of Antennas and Propagation (NJAP), 3*(2), 6-9.

[56] Stolfo, S. J., Fan, W., Lee, W., Prodromidis, A., & Chan, P. K. (2000). Cost-based modeling for fraud and intrusion detection: Results from the JAM project. *In IEEE Proceedings DARPA Information Survivability Conference and Exposition. DISCEX'00, 2,* 130-144.

[57] Talbi, E. G. (2009). *Metaheuristics: from design to implementation*. John Wiley & Sons.

[58] Tavallaee, M., Bagheri, E., Lu, W., & Ghorbani, A. A. (2009). A detailed analysis of the KDD CUP 99 data set. *In IEEE symposium on computational intelligence for security and defense applications, 1-6.*

[59] Yağız, E., Ozyilmaz, G., & Ozyilmaz, A. T. (2022). Optimization of graphite-mineral oil ratio with response surface methodology in glucose oxidase-based carbon paste electrode design. *Natural and Engineering Sciences, 7*(1), 22-33.

[60] Yang, J., Wang, L., & Shakya, S. (2022). Modelling Network Traffic and Exploiting Encrypted Packets to Detect Stepping-stone Intrusions. *Journal of Internet Services and Information Security, 12*(1), 2-25.

[61] Yang, X. S., & Deb, S. (2009). Cuckoo search via Lévy flights. *In IEEE World congress on nature & biologically inspired computing (NaBIC),* 210-214.

[62] Zhang, G., Wu, M., Duan, W., & Huang, X. (2018). Genetic algorithm based QoS perception routing protocol for VANETs. *Wireless Communications and Mobile Computing, 2018.*

[63] Zhou, Y., Cheng, G., Jiang, S., & Dai, M. (2020). Building an efficient intrusion detection system based on feature selection and ensemble classifier. *Computer networks*, *174*, 107247. https://doi.org/10.1016/j.comnet.2020.107247

[64] Zitouni, F., Harous, S., & Maamri, R. (2020). A distributed solution to the multi-robot task allocation problem using ant colony optimization and bat algorithm. *In Advances in Machine Learning and Computational Intelligence: Proceedings of ICMLCI*, 477-490. Singapore: Springer Singapore.

## Authors Biography

Dr. Mutasem Khalil Alsmadi is currently an associate professor at the Faculty of Applied Studies and Community Service, Department of Management of Information Systems, Imam Abdurrahman Bin Faisal University. He received his B.S. degree in Software engineering in 2006 from Philadelphia University, Jordan, his M.Sc. degree in intelligent systems in 2007 from University Utara Malaysia, Malaysia, and his Ph.D. in Computer Science from The National University of Malaysia. He has published more than one hundred papers in the image processing and algorithm optimization areas. His research interests include artificial intelligence, pattern recognition, algorithms optimization, and computer vision.

Dr. Rami Mustafa A Mohammad is an Associate Professor at the College of Computer Science and Information Technology at Imam Abdulrahman bin Faisal University. His research interests span a diverse range of fields, including information security, digital forensics, data mining, machine learning and web security. He has made notable contributions to the development of intelligent techniques for phishing website detection, intrusion detection systems, feature selection algorithms, and the application of machine learning to various security-related problems. Additionally, his research has addressed emerging challenges in domains such as blockchain, cloud computing, and the Internet of Things. Dr. Rami's research excellence has been widely recognized. For the past four years, he was among the Top Cited Researcher in his college, and he has been enlisted in the prestigious 'World Top 2% Scientists' list by Stanford University for two consecutive years, 2022 and 2023. These accolades underscore the profound impact and recognition of his scholarly work within the academic community.

Malek Alzaqebah obtained his B.Sc in computer science from Al-Balqa Applied University, Jordan and his master degree specializing in information technology from University Utara Malaysia (UUM). He did his Ph.D in computer science at University Kebangsaan Malaysia (UKM). Now, he is an assistant professor at the Department of Mathematics, Faculty of Science, Imam Abdulrahman Bin Faisal University, Dammam, Saudi Arabia. His research interest involves data mining and metaheuristic algorithms for combinatorial optimization problems such as university timetabling, job shop scheduling, vehicle routing and search-based software engineering.

Sana Jawarneh obtained her B.Sc in computer engineering from Yarmouk University and her Ph.D. in computer science at University Kebangsaan Malaysia. Now, she is an assistant professor at Department of computer science, The applied college, Imam Abdulrahman Bin Faisal University, Dammam, Saudi Arabia. Her research interest falls under meta-heuristic algorithms in various optimization problems.

Muath AlShaikh has been a Ph.D. holder in Computer Science since 2016, University of Bretagne Occidentale, France. He received his Master degree in computer science in 2010 from Utara University in Malaysia and his B.Sc in computer science in 2006 from AlBalqa University, Jordan. He is affiliated to Lab-STICC / UMR CNRS 6283, SFIIS team of the University of Bretagne Occidentale, France. He has been an Associate Professor with Computer Science Department, Saudi Electronic University, KSA. His research interests include Homomorphic, Cyber security, Watermarking, Cryptology, Information Security and Image processing and Computer vision.

Ahmad Al Smadi is an Assistant Professor at the Department of Data Science and Artificial Intelligence at Zarqa University. He completed his PhD degree in Computer Science and Technology from the School of Artificial Intelligence, Xidian University, Xi'an, China. He worked as a research assistant at the College of Technological Innovation, Zayed University, UAE. His research interests include information systems, computer vision, machine learning, and deep learning.

Fahad Ali Alghamdi is currently an assistant at the Faculty of Applied Studies and Community Service, Department of Management of Information Systems, Imam Abdurrahman Bin Faisal University, Saudi Arabia. He received his BS degree in computer science in 2002 from Imam Abdurrahman Bin Faisal University, Saudi Arabia, his MSc degree in computer science in 2009 from University Wollongong, Australia, and his Ph.D. in Computer Science in 2017 from Aberystwyth University in UK. His research interests include network, applied computer software, cloud computing and online learning. He is currently the Vice Dean of Faculty of Applied Studies and Community Service, Imam Abdurrahman Bin Faisal University, Saudi Arabia.

Jehad Saad Alqurni received the M.S. degree in computer science (software engineering) from the University of Wollongong, Wollongong, Australia, in 2009, and the Ph.D. degree in computer science from the Heriot-Watt University, U.K., in 2020. From 2005 to 2010, he was a Teaching Assistant withthe Imam Abdulrahman Bin Faisal University, Saudi Arabia, a Lecturer from 2010 to 2019, and has been an Assistant Professor since 2020. His research interests include the software engineering, human computer interaction, usability, web usability, E-Learning and information systems.

Hayat Alfagham received her BS degree in Computer Information Systems from King Faisal University, Saudi Arabia, her MSc degree in intelligent systems in 2015 from Kings College London, United Kingdom, and her Ph.D. in Computer Science in 2020 from Queen Mary University of London. She is currently an associate professor at the Faculty of Applied Studies and Community Service, Department of Management of Information Systems, Imam Abdurrahman Bin Faisal University. Her research interests include Link prediction, Big Data, Pattern recognition, Machine Learning, Network evolution.