IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

A Digital DNA Sequencing Engine for Ransomware Detection Using Machine Learning

Firoz Khan[1], Cornelius Ncube[2], and R.Lakshmana Kumar[3], Seifedine Kadry[4], Yunyoung Nam[5]

[1]Higher Colleges of Technology, Dubai, UAE (e-mail: fk7@hotmail.com)
[2]British University in Dubai, Dubai, UAE. (e-mail: cornelius.ncube@buid.ac.ae)
3Hindusthan College of Engineering and Technology, Coimbatore, India (e-mail: research.laksha@gmail.com)
[4]Department of Mathematics and Computer Science, Faculty of Science, Beirut Arab University, Beirut 11072809, Lebanon
[5]Department of Computer Science and Engineering, Soonchunhyang University, Asan 31538, South Korea

Corresponding author: Firoz Khan (e-mail: fk7@hotmail.com), and Yunyoung Nam (ynam@sch.ac.kr)

ABSTRACT

Malware is 'malicious software' programs that carry out many of the cyberattacks on the Internet, including cybercrime, fraud, scams and nation-state cyberwar. These malicious software programs come in a wide range of different classifications such as viruses, Trojans, worms, spyware, botnet malware, ransomware, Rootkit, etc. Ransomware is class of malware that holds the victim's data hostage by encrypting the data on a user's computer to make it unavailable to the user and only decrypt it after the user pays a ransom in the form of a sum of money. To avoid detection, different variants of ransomware utilise one or more techniques in their attack flow including Machine Learning (ML) algorithms. There is, therefore, a need to understand the techniques used ransomware development and their deployment strategy in order to understand their attack flow better to develop appropriate countermeasures. In this paper, we propose DNAact-Ran, A Digital DNA Sequencing Engine for Ransomware Detection Using Machine Learning. DNAact-Ran utilises Digital DNA sequencing design constraints and k-mer frequency vector. To measure the efficacy of the proposed approach, we evaluated DNAact-Run on 582 ransomware and 942 goodware instances to measure the performance of precision, recall, f-measure and accuracy. Compared to other methods, the evaluation results show that DNAact-Run can predict and detect ransomware accurately and effectively.

INDEX TERMS Ransomware, Digital DNA Sequence, Machine Learning, Active Learning

I. INTRODUCTION

The emergence of the Internet as a global technological-enabledplatform has resulted in a significantand exceptional increase in the use of the Internet-enabled devices, personal computers and digital applications that have enhanced human activitiesand significantly improved societal livesHowever, the adoption of the Internet as global technology platform has resulted in a significant rise in many cyber threats of different forms [1]. Malwareis one such malicious software program that has been responsible for a great deal of significant damage to networked systemsas well as significant cybercrime, fraud, scams and nation-state cyberwar activities [2].Figure 1 shows a typical malware attack flow [3].

Ransomware is one of the most widespread malware cyber-attack class that holds the victim's data hostage by surreptitiously encrypts the data on a user's computer to make it unavailable and only decrypts the data after the user pays a ransom in the form of a sum of money.

Ransomware generally comes in two primary forms: Crypto ransomware and Locker ransomware. The Crypto ransomware attacks the victim's machine by discreetly searching for documents in the victim's machine and then encrypting them. Victims can only get back access to their documents only after paying a ransom and the attackers providethe decryption keys.

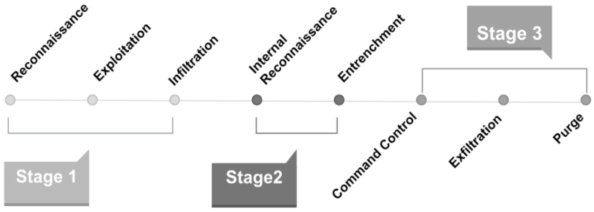


FIGURE 1. **A typical Malware Attack Flow (adapted from [3])**

Usually, crypto-ransomware do not encrypt the entire physical drive but targets user-generated files that have specific fileextensions such as .pdf, .jpg, and .doc files which typically contain valuable and personal user data [4]. On the other hand, the Locker ransomware-type locks the user's computer, using easy or complicated mechanisms, thus preventing the user from accessing their files [5]. A distinguishing feature of the Locker ransomware strain is that once the user’s computer and files have been taken hostage,the ransomware announces its presence to the userbydisplaying a ransom note on the user's computer screen demanding a ransom payment in exchange for the return of the user’s files [3, 4].Only after a successful payment, the useris grantedaccess back to the affected machine. CryptoWall is an example of locker ransomware that has been described as the most destructive ransomware threat on the internet as it is programmed to run on both 32-bit and 64-bit machines, thus increasing its chances to infect whatever machine it happens to be running on.WinLock is another early example of locker-ransomware, where a $10 ransom payment demandwas to be paid via SMS for the return of the locked files [5].

Even though Ransomware continues to evolve, and more sophisticated variant strains are being deployed all the time, all follow a typical six-stage attack flow kill-chain [6]:

Stage 1. Distribution Campaign – in which attackers surreptitiously trick victims to download a dropper code that initiates the infection. Conventional techniques used for this stage are email phishing, social engineering weaponised websites or USBferry [7]

Stage 2. Malicious Code Infection– in which executable malicious code is downloaded and the ransomware installs itself on the victim’s machine

Stage 3. Malicious Payload Staging – the ransomware sets up, embeds itself in a system, and establishes persistency to exist beyond a reboot. The malicious code establishes connectivity with its command and control server that is controlled by the attackers.

Stage 4. Scanning – the ransomware scans the victim’s computer and network accessible resources for files to encrypt.

Stage 5. Encryption – once all the files have been discovered, encryption begins, and all files are encrypted

Stage 6. Payday – at this stage, the entire victim’s data is gone, a ransom note is generated and displayed to the victim’s screen demanding payment, and the attacker waits to collect on the ransom. The victim is then consistently pressurised to pay the ransom

The most commonly preferred encryption technique used in most ransomware strains is usually symmetric encryption due to its efficiency and speed. However, some ransomware strains use hybrid asymmetric and symmetric encryption methodologies.The hybrid methodologies involve the use of an asymmetric key for encrypting the victims’ files and an asymmetric public key to encrypt the symmetric key. A point to note is that the infected users are being pressurised to make ransom payments by introducing social engineering techniques into the attack. Examples of such techniques include the impersonation messages that are coming from law enforcement agencies,and users need to pay fines for committing crimes using the computer that may include downloading music and movie files illegally.

Several techniques are frequently used by cybercriminals to install ransomware on a user's computer, including Phishing or spam emails, exploit kits, downloader and Trojan botnets, social engineering tactics and Traffic distribution system.

Ransomware operation varies according to the family of ransomware. The operation can be; denied access to data files of users by encrypting them or taking over the boot process of a system and disable access to the system entirely. Ransomware is a subcategory under malware, and there are several behavioural differences between malware and ransomware [8]. A significant difference is that malware aims to remain hidden from the user for as long as possible, to continue its functions in a stealth mode. Whereas, the primary purpose of ransomware is to be shown to the user and make him aware of the infection. WannaCry ransomware is the most recent successful ransomware attack. The ransomware exploited the vulnerability named EternalBlue present in the SMB protocol of Windows systems and infected multitudes of users all over the world. User data was encrypted, and bitcoin payments were demanded in exchange for the decryption key [9].

II. MOTIVATION

The rise in alarge number of ransomware attacks has prompted governments, organisations and users to secure and create backups of their critical data. However, due to the highly profitable nature of these cyberattacks, the newer ransomware strains are continuallyevolving, and attackers continue to create more new sophisticated ransomware every day. The current defence mechanisms to detect, analyse and defend against ransomware are not effective enough and are unable to cope with the volumeof attacks. The primary motivation for this paper is to propose an efficient and unique Digital DNA Sequencing Engine that usesthe ML algorithm to detect ransomware before the initial attack stage takes place.

The paper proposes DNAact-Ran, an ML-based digital DNA sequencing engine for detecting and classifying ransomware through sequencing its digital DNA using an active ML approach. DNAact-Ran first selects key features from the preprocessed data using Multi-Objective Grey Wolf Optimization (MOGWO) and Binary Cuckoo Search (BCS) algorithms. Thereafter the digital DNA sequence is generated for the selected features using the design constraints of DNA sequence and k-mer frequency vector.

III. INNOVATIVE CONTRIBUTION

Most current ransomware detection techniques are based onbehaviour analysis of the malware. However, developing detection signatures requires acquiring behaviours in the first place. This then implies that a ‘first attack’ has to be successful in order to get the valuable information needed to develop detection signatures. However, in order to avoid detection, malware developers use obfuscation techniques such as binary code packing [10][11]. Code packing is the technique of encrypting the original code including the data and restoration routine function with the packed program itself. Then when the packed program is executed, the restoration routine code restores the original code and data to its original form [11]. An even more challenging code obfuscation techniques are the layered polymorphic malware that mutates their code as well as their decryption method and only reveals a portion of the code at any execution stage [12] and metamorphic malware that mutates their code in its decrypted form resulting in different malware strand in each new mutation [13].

However, to avoid detection, malware developers are adapting their obfuscation methods to detect unpacking tools or to by-pass the unpacking tool process by tracking their unpacking methods [14]Hiding the function code responsible for malicious behaviour makes it difficult for current signature-based behaviour analysis anti-malware engines to detect the malicious code and the malware's functionality [11][14].In this paper, we argue that signature-based ransomware detection engines are no longer effective. We, therefore, propose a revolutionary Digital DNA sequencing engine that uses Machine Learning algorithms to characterise ransomware based on their 'digital genotypes and inferred digital phenotypes' in order to identify their malicious functions [15].

Two main innovative claims of the proposed digital DNA sequence generation approach are the high accuracy and reliability in digital DNA sequencing and an active Machine Learning algorithm that uses a set of 'genome rules' to classify software programs and data as either ransomware or goodware.

The paper makes the following contributions:

- A new method that uses Digital DNA Sequencing Engine for Ransomware Analysis using an AI Machine Learning Network to detect and classify ransomware;
- The use of MOGWO and BCS algorithms to generate Digital DNA Sequences computation;
- digital DNA sequence constraints and k-mer frequency vector based on the DNA sequences;
- A software product model, a requirement model and compliance model that models compliance and correlation relationships for ransomware detection;
- A Classification methodology that uniquely detects and classifies the detected ransomware into well-known familiesbased on their 'digital genome' [9][16][17];
- A concept demonstrator tool that successfully detects and classify ransomware using an active machinelearning algorithmand real-world datasets to demonstrate the feasibility of the above concepts

The rest of the paper is organised as follows. Section 2 briefly describes related work in ransomware detection. Section 3 details the proposed DNAact-Ran approach based on AI-ML and DNA sequences. The experimental results are evaluated in Section 4, and finally, section 5 concludes the paper and outlines future work.

IV. RELATED WORK

Ransomware attacks have been increasing throughout recent years, and many methods have been proposed to detect and prevent them. Most current ransomware detection and analysis methods fall into two main categories -dynamic orstatic approaches. Dynamic analysis method involves creating an isolated environment and running the malware within that environment to recognise its functional behaviour. Static approaches, on the other hand,include reverse engineering the malicious code to understand the working of the malware and then to developdefences against it.

Sgandurra et al. [4] proposedEldeRantool that checks characteristic ransomware signatures by analysing a set of actions during the initial phases of the attack flow kill-chain (i.e. Stage 2). EldeRan dynamically detects and classify ransomware by analysing operation activities such as registry key operations, calls from Windows APIs, folder and file system operations. EldeRanuses Logical Regression classifier algorithm an ML algorithm, to classify each user application, and has additional functionality to identify and create signatures for as yet unknown ransomware.

Andronio et al. [18] proposed the HelDroidsystem that detects a class of ransomware that is designed to target Android platforms.The HelDroidsystem uses code characteristics, including application manifests and call functions to identify ransomware and its family class using Natural Language Processing (NLP). The HelDroid system is trained to identify common messages that appear in the ransomware code to identify it. However,HelDroid's main weakness is that it uses a text classifier to detect and characteriseransomeware[18].Mercaldo et al. [19],developed a parser that automatically identifies related ransomware instructions in a three-step process by analysing sample code and detecting the associated Android ransomware family.

Kharraz et al. [20] proposed UNVEIL, which is a dynamic analysis tool that checks three elements: file system activities, access patterns and I/O data buffer entropy.UNVEIL also has text analysis techniques for detecting screen lockers and threatening ransom notes demanding payment that are similar to those proposed in [20].

Song et al. [21]proposed a method to detect and prevent modified ransomware from attacking Android platforms. The proposed method has a very high and fast detection rate as the tool is designed in such a way to be embeddedwithin theandroid source code rather than as an external mobile application. This makes it a very powerful techniqueas it can detect theransomwareand its variants even if it does not have its signature pattern by monitoring. Its key is the processor, memory usage and I/O rates are to detect abnormal behaviours. If any discrepancy is detected, the system takesimmediate action to stop the process anddeletes the program associated with the process [22].

Aragorn et al. [23]developed a tool that uses deep learning techniques to detect ransomware. A deep neural network was developed and used to train the perceptron with critical payloads selected from data packets extracted from real network traffic. This method can detect many ransomware variants includingzero-day exploits.

Shaukat and Ribeiro [24] proposed RansomWall, a layeredtool that was designed toprotect against crypto-ransomware after analysing a vast dataset of ransomware families. RansomWall combines dynamic and static analysis methodologies to produce a unique compact set that detects the ransomware behaviourpatterns. RamsomWall is also designed todetect zero-day ransomware exploits and its Strong Trap Layer function can detect ransomware attack at the early stages of the kill-chain. This is done by detecting and classifying any suspicious activity in the initial layers. If any files are found to have been modified by the rogue process, the files are backed up to protect the user's data until the rogue process has beendetermined to be eitherdangerous or not.

There are many Ransomware Detection Methods that employ different techniques such as Signature-based Detection, Honeypot, Hashing, Shannon's Entropy and Machine Learning. Each technique has advantages and disadvantages.

Most current antivirus tools are signature-based and detect ransomware through matching binary patterns and monitoring APIs. However, if ransomware changes their behaviour or uses packers to camouflage malicious code, the antivirus tools would not be able to detect them without an updated signature [25]. That is, the antivirus software cannot defend effectively against new and unique attacks. Also, signature-based tools cannot detect that ransomware attack in its early stages of the kill-chain, (e.g. Stage 2).

Through an extensive analysis of the different methods, we choose MLas the best option in our proposed approach. In this paper, we propose DNAact-Ran, an approach that uses ML to detect ransomware by sequencing its digital DNA. The following section briefly describes how DNAact-Ran works and provides its underlying architecture.

The aim of DNA sequence design is done by satisfying constraints to avoid such unexpected molecular reactions and is considered to be an approach of control. Good DNA sequences are designed by using constraints such as Continuity, H-measure, GC content and Melting Temperature, among others

IV. HOW DNAACT-RAN WORKS

This section introduces the proposed DNAact-Ran approach and briefly describes how it detects the ransomware.

To detect if a program is ransomware or goodware, DNAact-Ran first, selects significant features using MOGWO and BCS algorithms. Next, it generates the digital DNA Sequence for selected features, and finally, it classifies the instances as either goodware or ransomware using active learning concept. Figure 2 shows the proposed architectureDNAact-RanDNAact-Ran approach detects ransomware in three key process steps. Feature Selection, DNA Sequence Generation and Ransomware Detection.

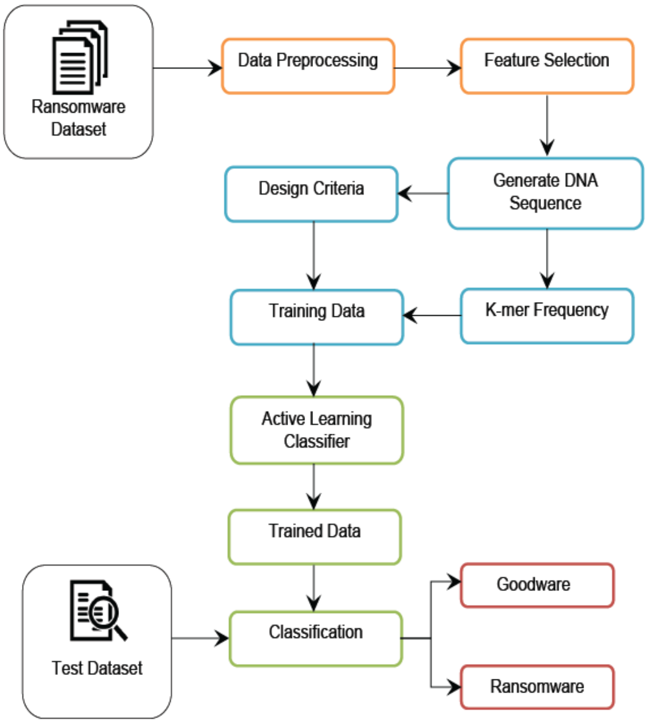


FIGURE 2. **DNAact-Ran Architecture**

Each step is described next:

A. FEATURE SELECTION

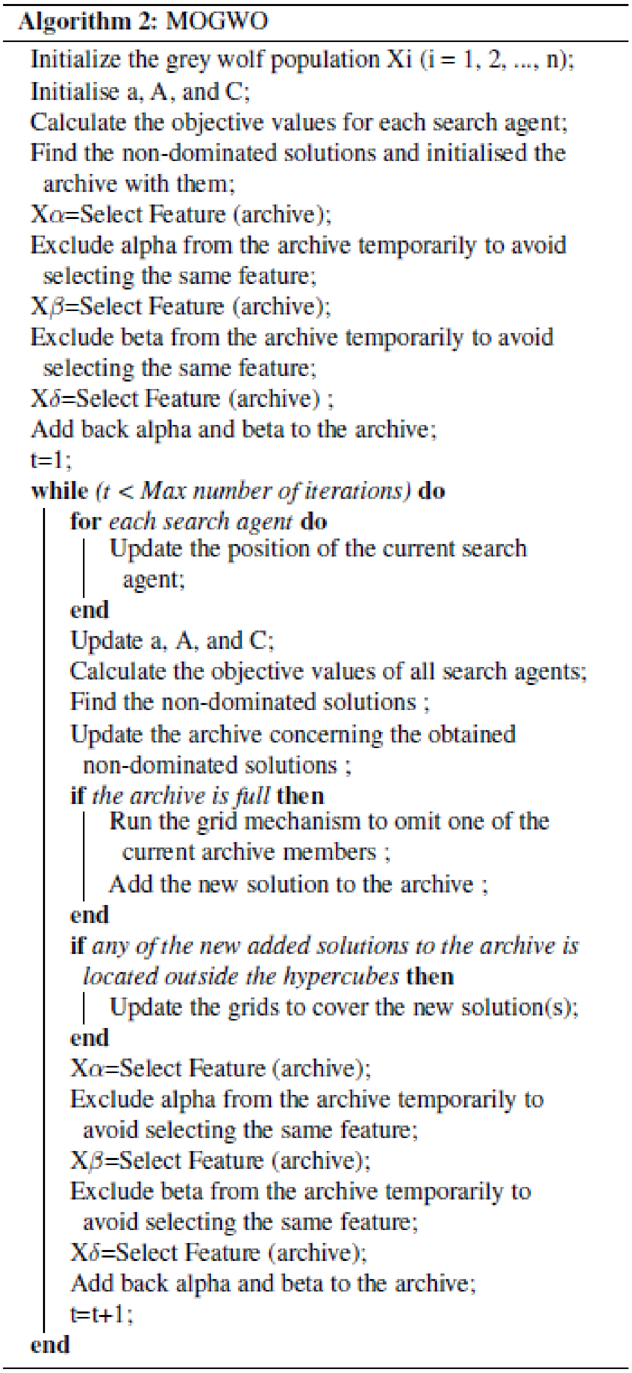
Data Preprocessing is a data mining technique for converting raw data into an understandable user format. Real-world data usually has missing values, consists of noisy values, and is generally incomplete, inconsistent and include outlier information. Hence, raw data need to go through a preprocessing step before being mined for useful information, and this process step enhances and ensures data efficiency. Feature selection is one of the essential processes in ML. Feature selection is used to remove irrelevant features and reduces storage and computational cost. Searching for the best set of features is a challenging and complex task due to the vast search space in scenarios with an extensive

Algorithm 1: Proposed Feature Selection

```
Read the Dataset D;
D1 = Remove missing value records in D;
D2 = Remove features (columns) that contains zero
  value for all records in D1;
F1 = Apply MOGWO;
F2 = Apply BCS;
SF = F1 ∩ F2;
Create new Dataset D' using SF (Selected Features);
```

number of features. DNAact-RANuses MOGWO and BCS to select the relevant features from the collected dataset. Algorithm 1 below shows the pseudocode for selecting feature.

Mirjalili et al. [25] proposed GWO,a swarm intelligence based algorithm. Grey wolves inspire the GWO algorithm that searches for the most optimal method for hunting preys. MOGWO is an extension of GWO.

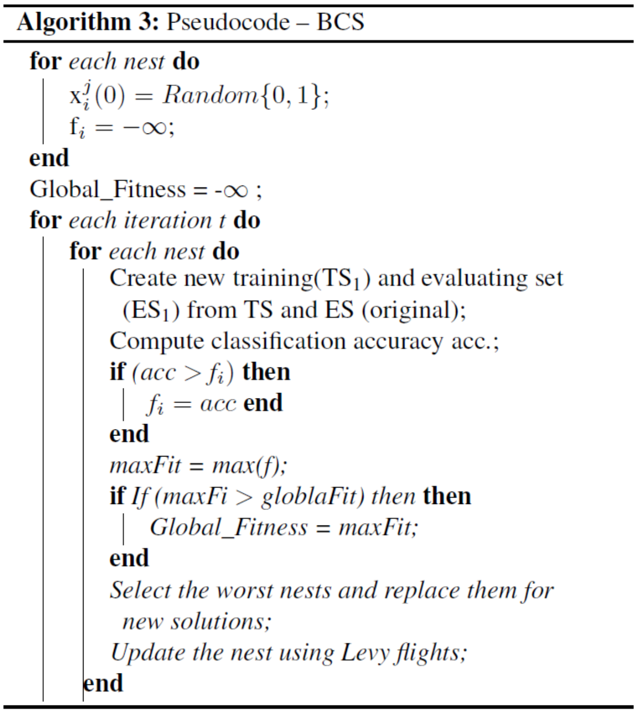


```
Algorithm 2: MOGWO
Initialize the grey wolf population Xi (i = 1, 2, ..., n);
Initialise a, A, and C;
Calculate the objective values for each search agent;
Find the non-dominated solutions and initialised the
  archive with them;
Xα=Select Feature (archive);
Exclude alpha from the archive temporarily to avoid
  selecting the same feature;
Xβ=Select Feature (archive);
Exclude beta from the archive temporarily to avoid
  selecting the same feature;
Xδ=Select Feature (archive) ;
Add back alpha and beta to the archive;
t=1;
while (t < Max number of iterations) do
    for each search agent do
        Update the position of the current search
          agent;
    end
    Update a, A, and C;
    Calculate the objective values of all search agents;
    Find the non-dominated solutions ;
    Update the archive concerning the obtained
      non-dominated solutions ;
    if the archive is full then
        Run the grid mechanism to omit one of the
          current archive members ;
        Add the new solution to the archive ;
    end
    if any of the new added solutions to the archive is
      located outside the hypercubes then
        Update the grids to cover the new solution(s);
    end
    Xα=Select Feature (archive);
    Exclude alpha from the archive temporarily to
      avoid selecting the same feature;
    Xβ=Select Feature (archive);
    Exclude beta from the archive temporarily to
      avoid selecting the same feature;
    Xδ=Select Feature (archive);
    Add back alpha and beta to the archive;
    t=t+1;
end
```

MOGWO consists of two main components, a grid and an archive. The responsibility of the grid component is to keep the archive solutions as varied as possible. In MOGWO, the objective space is divided into several regions named grids. All grid locations need to be recalculated if a newly obtained solution lies outside the gridto take into account of the new solution. Moreover, a new solution is directed to the portion of the grid with the lowest number of particles, if the solution lies within the grid. The archive component decides whether a solution should be added to the archive or ignored. A new solution will be immediately discarded if the solution is dominated by one of the archive members. Alternatively, if the archive members do not dominate the new solution, the solution would be added to the archive. If a new solution dominates an existing member of the archive, the old member would be replaced by the new solution. Algorithm 2 shows the pseudocode for MOGWO.

Yang and Deb [26] proposed the Cuckoo Search (CS) which is a heuristic search algorithm as an algorithm. The basis of the CS algorithm is adopted from the reproduction strategy of the wild cuckoo birds. A feature selection model based on a binary version of the Cuckoo Search (BCS)consists of a search space modelled as a d-cube. In this model, *d* refers to the number of features. A set of binary coordinates is connected with each nest that denotes whether a feature will belong to the final set of features.

Additionally, the supervised classifier's accuracy determines the function to be maximised. Algorithm 3 shows the pseudocode for the BCS Feature Selection.DNAact-Ran, feature selection method, selects the most relevant features.



```
Algorithm 3: Pseudocode – BCS
for each nest do
    $x_i^j(0) = Random\{0, 1\}$;
    $f_i = -\infty$;
end
Global_Fitness = -∞ ;
for each iteration t do
    for each nest do
        Create new training(TS$_1$) and evaluating set
          (ES$_1$) from TS and ES (original);
        Compute classification accuracy acc.;
        if (acc > $f_i$) then
            $f_i = acc$ end
        end
        maxFit = max(f);
        if If (maxFi > globlaFit) then then
            Global_Fitness = maxFit;
        end
        Select the worst nests and replace them for
          new solutions;
        Update the nest using Levy flights;
    end
```

B. DIGITAL DNA SEQUENCE GENERATION

DNA is the biological blueprint used for building proteins and other cellular components of living organisms. It is comprised of a long stretch of adenine (A), guanine (G), cytosine (C), and thymine (T) molecules, commonly referred to as "bases" due to their chemical nature. They are also referred to as nucleotides. DNA is represented computationally by character strings containing only the characters A, G, C and T [27].

This section describes how DNAact-RANgenerates a digital DNA sequence of ransomware. After a feature selection as described above, a newly generated dataset is used to generate the digital DNA sequence. The design constraint of Digital DNA is then computed, and k-mer frequency vector is generated for the DNA sequence. Based on these computations and vector, a new dataset is generated for ransomware detection training phase. A synthetic DNA representation of a digital artefact is represented as a sequence of DNA characters (A, C, G and T).It is considered synthetic as it represents the content of a digital artefact rather than biological DNA. Pedersen et al. [27]and Xiao et al. [28]used synthetic DNA bycreating a reversible translation of the byte sequence of a digital artefact. A digital artefact can be considered as a sequence of byte values, with each byte being a sequence of four two-bit pairs. Each two-bit pair has four possible values 00, 01, 10 and 11 which were mapped to the four biological DNA characters A, T, G and C. This mapping procedure generates four digital DNA characters for each byte in the digital artefact. DNAact-Ran generates ransomware digital DNA Sequence using the following mapping procedure shown in Table 1 below:

TABLE I
DNA CHARACTER MAPPING

Binary Bit	DNA Character
00	A
01	C
10	G
11	T

Consider the sample binary features ‘0 1 0 0 0 1 0 1 0 1 0’. 'CGAACGCGCG' is, therefore, the corresponding biological DNA encoding for the sample this sample binary feature.

The DNA sequence design is an approach of the control, which aims to design DNA sequences, satisfying constraints to avoid such unexpected molecular reactions. Generally,a good DNA sequences design should consider constraintssuch as H-measure, continuity, melting temperature, GC content, among others [29]. DNAact-Ran uses the following three constraints for digital DNA Sequence design:

Tm Constraint: uses a simple method that assigns 4°C to each G-C pair and 2°C to each A-T pair. Tm is the sum of these values for all individual pairs in a DNA double-strand. The process takes into account that the A-T bond is stronger than the G-C bond. The following formula is used to compute Tm.

$$Tm = 4°C(G+C)+2°C(A+T) \qquad (1)$$

GC Content Constraint: The percentage of G and C in a DNA sequence is known as the GC Content (GCC). GC content is usually calculated as a percentage value and sometimes referred to as GC-ratio or G+C ratio. The formula for calculating the GC-content percentage is shown below:

$$GCC = \frac{G+C}{A+T+G+C} * 100\% \qquad (2)$$

AT_GC Ratio Constraint: This ratio is calculated as follows:

$$(A+T)/(G+C) \qquad (3)$$

The DNA Sequence constraints for the sequence ‘CGAACGCGCG’ is therefore Tm= 36, GCC= 80 and AT_GC=0.25.

K-mers are sub-sequences of length *k* contained within a biological sequence and is used in the field of bioinformatics. K-mers are composed of nucleotides bases and are used in the field of genome and sequence analysis to assemble DNA sequences. Typically, the term k-mer refers to all of a sequence's subsequences of length k such that the sequence AGAT would have four monomers (A, G, A, and T), three 2-mers (AG, GA, AT), two 3-mers (AGA and GAT) and one 4-mer (AGAT).

The feature vector $F_k(s)$ for an input digital DNA sequences was constructed from the number of occurrences of all 4^k possible k-mers (given the nucleotide alphabet {A, C, G, T}), divided by the total length of s. Any ambiguous nucleotide codes (e.g., 'N' for completely ambiguous nucleotides) were removed from s before computing $F_k(s)$. As a concrete example, suppose s= CGAACGCGCG and k= 2. Then, if we use the arbitrary order [AA, AC, AG, AT, CA, CC, CG, CT, GA, GC, GG, GT, TA, TC, TG, TT] for 2-mers, the k-mer frequency vector for s is [1, 1, 0, 0, 0, 0, 4, 0, 1, 2, 0, 0, 0, 0, 0, 0] and thus

$$F_k(s) = [0.1, 0.1, 0, 0, 0, 0, 0.4, 0, 0.1, 0.2, 0, 0, 0, 0, 0, 0]$$

Using these design constraints and k-mer frequency, the new dataset is formed for further process.

C. RANSOMWARE DETECTION

This section briefly describes how the active learning algorithm detects ransomware. The training dataset was generated based on the selected features with DNA design constraints and k-mer frequency. The dataset was trained using active learning classifier. Digital DNA sequences are then randomly generated from test data. The active learning algorithm also computes DNA constraints and k-mer frequency. The test data is classified as goodware or ransomware using an active learning classification algorithm.Finally, the ransomware familyis detectedusing a various classification algorithm.

The most significant issue with many ML applications is the effort and time required to annotate large quantities of data sets that are required for supervised learning in the process of training a high-accuracy classifier. To solve this issue, a protocol called active learning has been proposed and designed. The active learning protocol decreases the cost by identifying highly informative data points to be annotated sequentially and to be used by the learning algorithm.

Four categories of active learning techniques are usually performed[29]:

- Query strategies based on uncertainties -where instances with the lowest prediction confidence are queried;
- Query strategies based on disagreement -which queries the instances on which the hypothesis space has the most disagreement degree on their predictions;
- Minimise the expected variances and error by labelling the instances on the pool of unlabelled instances;
- Exploiting the structure information among the instances

As the proposed active learning algorithm needs a linear regression classification for its initial prediction, we firstly present the underlying linear regression model.

To train a Linear Regression algorithm, first, the regression task is adapted into the supervised ML.Then the Regression model is used to detect the relations between forecasting and variables. Relationships between independent and dependent variables influence the type of regression models to be used to address the issue on hand. Additionally, the regression model used is based on the number of independent variables.

Finally, the Linear regression algorithm is used to predict a quantitative response Y from the predictor variable X. A simple linear regression model is shown below.

$$Y = \beta_0 + \beta_1 X + \varepsilon \qquad (4)$$

Where Y is, the study or dependent variable andX is as an explanatory or independent variable.; β_0 and β_1 are theregression coefficients, (i.e. parameters of the model,) Whereβ_0 is the intercept parameter and β_1 is the slope parameter; component ε is an unobservable error which accounts for the failure of data to stay on the straight line and characterises the variance between the true and detected realisation of Y. The effect of all deleted variables in the model is one of several reasons for such difference. The variables may be inherent randomness in the observations or qualitative.

The model aims to predict Y value by achieving the best-fit regression line. Also, the error difference between the true value and predicted value needs to be minimum. Therefore, the β_0 andβ_1 values need to be updated to reach the best value and minimise the error between true Y value (Y) and predicted Y value (pred).

Root Mean Squared Error (RMSE) between true Y value (Y) and predicted Y value (pred) is the Cost function (J) of Linear Regression using the formula below.

$$J = \frac{1}{n} + \sum_{i=1}^{n} (pred_i - Y_i)^2 \qquad (5)$$

A sequence of training instances is used by a learner regression algorithm to iteratively learn $\{(X_t, Y_t) \mid t = 1,\ldots,T\}$, where and $Y_t \varepsilon$-1, +1 is its true class label. The goal of online binary classific is its true class label $X_t \varepsilon \Re^d$is the feature vector of the t-th instance. The objective of online binary classification is to learn a linear classifier,

$$Y_t = pre(W_t^T X_t) \qquad (6)$$

Where $W_t \in \Re^d$ is the weight vector at the t-th round.

When X_t is received, an online active learning algorithm needs to decide whether to query the true label Y_t or not, which is not the same as regular online supervised learning. An external expert will be asked to provide the true label if the algorithm chooses to request the true label. The algorithm may undergo some positive loss and implement traditional online learning techniques to update the model W_t, once the true label is observed.

If not met, the instance will be ignored by the algorithm and continue to process the next one. Algorithm 4 below shows the pseudocode of the active learning algorithm for the training dataset.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3003785, IEEE Access

Algorithm 4: Active Learning Algorithm

```
Initialize Learning Rate LR, Regularization Parameter
  RP and Smoothing Parameter SP;
Read Train dataset Tr;
for each instance (inst) in Tr do
    pre= Predict the class value of inst using linear
      regression model;
    if (pre > 0) then
        Ypre = 1;
    else
        Ypre = -1;
    end
    Compute r=1/RP , v=1/pre1, c=0.5*(-LR/r+v);
    Compute p=Abs(pre)+c;
    if (p > 0) then
        Compute sm=SP/(SP+p);
        if (random (0, 1) < sm) then
            Zt=true
        end
    else
        Zt=false;
    end
    if (Zt == true) then
        Set inst class value as Ypre;
    end
end
```

The training data is well trained using active online learning. DNA sequences are randomly generated and selected for test data. For each test DNA sequences, the design constraints and k-mer frequency are computed. The test data is predicted using a linear regression model based on active learning training data.

After learning the DNA sequence in ransomware, the family of the ransomware needs to be analysed. The ransomware families are identified by the codes, shown in Table 2.

TABLE II
RANSOMWARE FAMILIES

Family	ID
Goodware	0
Citroni	1
CryptLocker	2
CryptoWall	3
KOLLAH	4
Koyter	5
Locker	6
MATSNU	7
PGPCODER	8
Reveton	9
TeslaCrypt	10
Trojan-Ransom	11

After classifying ransomware into families, the ML algorithms Random Forest, Sequential Minimal Optimization and Naïve Bayes, are used to analyse the families.

V. EXPERIMENTAL RESULTS

This section presents the experimental results and validation to assess the accuracy of DNAact-Ran in detecting ransomware by measuring the performance of the classifier compared to other ML techniques. These experiments were performed using Java (version 1.8). The real-world dataset was obtained from https://github.com/PSJoshi/Notes/wiki/Datasets. The dataset contains 1524 records and 30970 features of which 582are ransomware and 942 are goodware applications. In this dataset, 300 records and 16383 features with 150 ransomware and 150 goodware applications were used. The following sections describe the evaluation of feature selection forransomware prediction.

A. EVALUATION OF FEATURE SELECTION

DNAact-Ran applies preprocessing and feature selection method before the detection of ransomware. The objective of feature selection is to reduce the number of features, to eliminate irrelevant, noisy and redundant features and to select the most representative features. MOGWO and BCS techniques are used to select significant relevant representative features.

Initially, the dataset contained 16383 features; after applying preprocessing step, it wasreduced 426 features. Then top 26 significant features were selected using the MOGWO and BCS feature selection algorithm. Figure 3 shows the feature selection comparison of MOGWO, BCS and suggestedDNAact-Ran method.

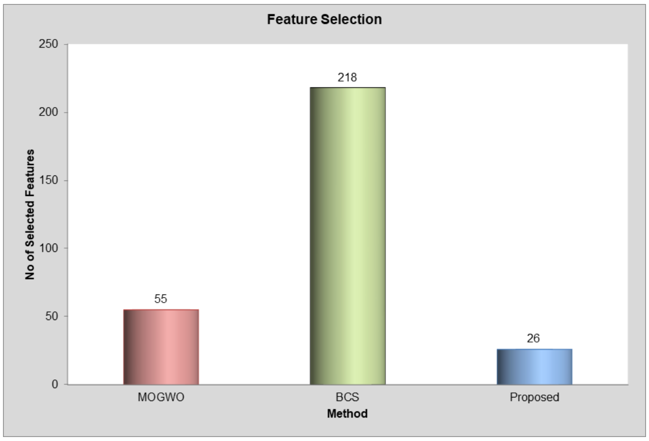


FIGURE 3. **Feature Selection Comparison**

Compared to the cuckoo search method, MOGWO and proposed methods reduce more features. The proposed method selects the features based on the intersection of MOGWO and cuckoo search. Only common features are selected for the next process.

Figure 4 shows the feature selection execution time comparison of MOGWO and BCS. MOGWO takes less time compared to BCS.

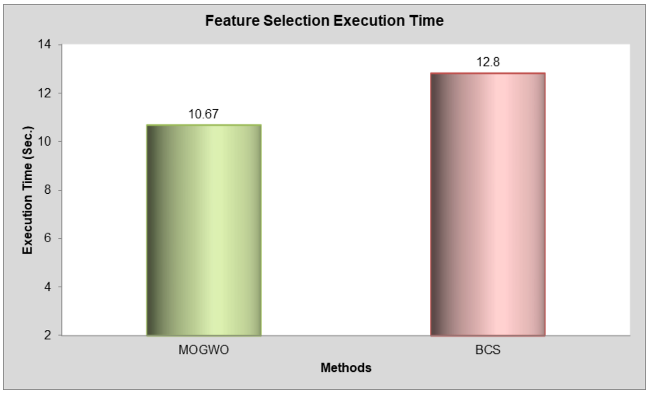


FIGURE 4. Execution Time Comparison

B. EVALUATION OF RANSOMWARE DETECTION

The following metrics were used to evaluate active learning-based ransomware detection:: precision, recall, f-measure and accuracy. DNAact-Ran classification accuracy is compared to the traditional ML classifiers (Naïve Bayes, Decision stump and AdaBoost). For the active learning algorithm, three main parameters are used: Learning Rate (LR), Regularization Parameter (RP) and Smoothing Parameter (SP). The value for these parameters is initial set as LR=10, RP=0.5 and SP=0.6. Figure 5 shows the accuracy comparison of the proposed active learning algorithm with Naïve Bayes, Decision Stump and AdaBoost. From the figure, it can be shown that the proposed detection algorithm gives better accuracy compared to other algorithms.

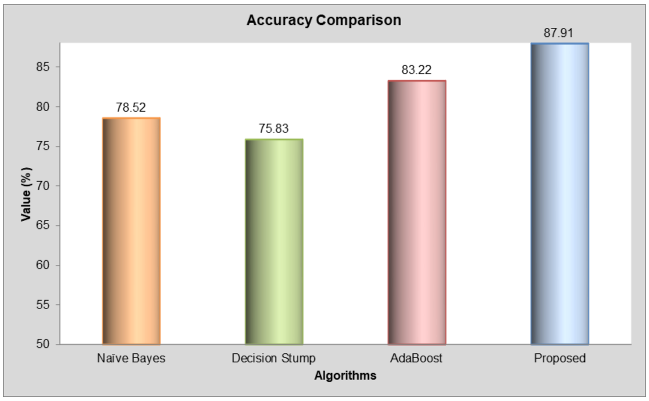


FIGURE 5. Accuracy Comparison

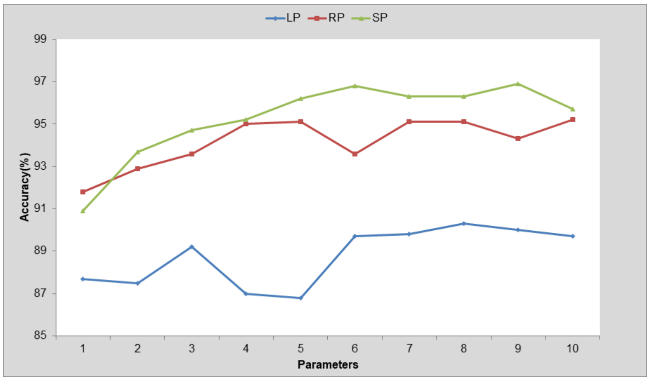


FIGURE 6. Accuracy Comparison of different parameters

The results are based on the Learning Parameter, Regularization Parameter and Smoothing Parameter. To improve the performance, automatically readjust these parameters for achieving a good result. Figure 6 shows the accuracy comparison of various parameters.

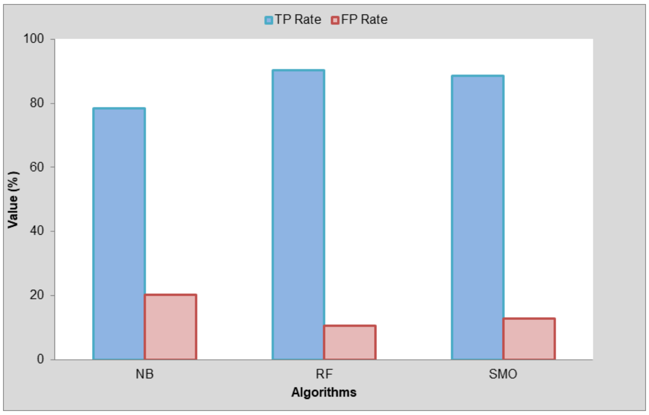


FIGURE 7. TP and FP Rate Comparison

After learning or detecting the program is ransomware, it then classifies what type is the ransomware. The classification algorithms Naïve Bayes (NB), Random Forest (RF) and Sequential Minimal Optimization (SMO) are used to analyse the ransomware family.

Figure 7 shows the True Positive (TP) and False Positive (FP) rate for three classification Algorithms. RF and SMO give less (/more) FP (/TP) rate compared to NB.

The evaluation metrics for the classification algorithms are shown in figure 8 and show that the RF algorithm gives better results compared to SMO and NB.

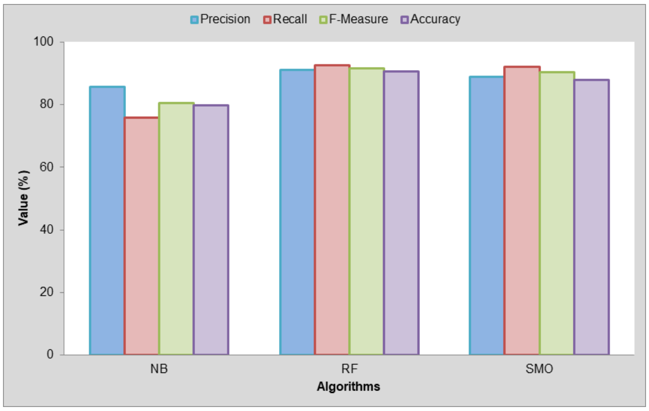


FIGURE 8. Evaluation Metrics Comparison

VI. CONCLUSION AND FUTURE WORK

In this paper, we proposed a DNAact-Ran for ransomware detection method. The ML algorithm was effectively applied for ransomware detection. The real-time dataset was used to validate the effectiveness and efficiency of the proposedDNAact-Ran method. A set of evaluation measures were used to evaluate the proposed DNAact-Ran. The proposed methodwas compared to several existing ML algorithms.

The proposed active learning algorithm was compared to Naïve Bayes, Decision Stump and AdaBoost classification algorithms. The experiment results show a 78.5% detection accuracy for Naïve Bayes, 75.8\% for Decision Stump, 83.2% for AdaBoost and 87.9% for the proposed active learning algorithm. The experiment partially proves that active learning classifiers are better at efficiently detecting ransomware.

ACKNOWLEGMENT

This research was supported by Korea Institute for Advancement of Technology(KIAT) grant funded by the Korea Government(MOTIE) (P0012724, The Competency Development Program for Industry Specialist) and the Soonchunhyang University Research Fund.

REFERENCES

[1] L. Xue and G.Sun, "Design and implementation of a malware detection system based on network behaviour", *Security and Communication Networks*, vol. 8, no. 3, pp. 459-470, 2015

[2] Y.Fan, Y.Ye and L.Chen, "Malicious sequential pattern mining for automatic malware detection", *Expert Systems with Applications*, vol. 52, pp.16-25, 2016

[3] MITRE, “Att&ck knowledge base.” [Online]. Available: https://attack.mitre.org

[4] D.Sgandurra, L.Munoz-Gonzalez, R.Mohsen and E.C.Lupu, "Automated dynamic analysis of ransomware: Benefits, limitations and use for detection", *CoRR*, (preprint), 2016. [Online]. Available: http://arxiv.org/abs/1609.03020

[5] O.M.K.Alhawi, J.Baldwin and A.Dehghantanha, "Leveraging Machine Learning Techniques for Windows Ransomware Network Traffic Detection", *Advances in Information Security*, vol. 70, pp. 1–11, 2018

[6] The Anatomy of a Ransomware Attack (2016), The Threat Research Report, [Online]. Available: http://www.exabeam.com,

[7] Joey Chen, (2020), Tropic Trooper’s Back: USBferry Attack Targets Air-gapped Environments. [Online]. Available: https://documents.trendmicro.com/assets/Tech-Brief-Tropic-Trooper-s-Back-USBferry-Attack-Targets-Air-gapped-Environments.pdf

[8] K.Gangwar, S.Mohanty and A.Mohapatra, "Analysis and Detection of Ransomware Through Its Delivery Methods", *Proceedings of the International Conference on Recent Developments in Science Engineering and Technology*, pp. 353–362, 2017

[9] M.Mimoso, "Leaked NSA Exploit Spreading Ransomware World Wide", [Online]. Available: https://threatpost.com/leaked-nsa-exploit-spreading-ransomware-worldwide/125654/ ,2017

[10] Chandan Kumar Behera and D. LalithaBhaskari (2015), Different Obfuscation Techniques for Code Protection, *4thInternational Conference on Eco-friendly Computing and Communication Systems*, / Procedia Computer Science 70 (2015) 757 – 763

[11] Roundy, K. A. and Miller, B. P. 2013. Binary-code obfuscations in prevalent packer tools. *ACM Comput. Surv. 46, 1, Article 4* (October 2013)

[12] Philippe Beaucamps, “Advanced Polymorphic Techniques,” *International Journal of Computer Science*, vol. 2, no. 3, pp. 194-205, (2007).

[13] BabakBashari Rad, Maslin Masrom, Suhaimi Ibrahim (2012), Camouflage in Malware: from Encryption to Metamorphism, *IJCSNS International Journal of Computer Science and Network Security*, VOL.12 No.8, August 2012

[14] Kevin Coogan, Saumya K. Debray, TasneemKaochar, Gregg M. Townsend (2009), Automatic Static Unpacking of Malware Binaries, *16th Working Conference on Reverse Engineering*, WCRE 2009, 13-16 October 2009, Lille, France

[15] Fortuna MA, Zaman L, Ofria C, Wagner A (2017), The genotype-phenotype map of an evolving digital organism, *PLoSComputBiol* 13(2)

[16] N. Udayakumar, V. J. Saglani, A. V. Cupta and T. Subbulakshmi, (2018), Malware Classification Using Machine Learning Algorithms, *2nd International Conference on Trends in Electronics and Informatics (ICOEI),* Tirunelveli, 2018, pp. 1-9

[17] HanqiZhangab, Xi Xiaob, Francesco Mercaldoc, Shiguang Nib, FabioMartinellic and Arun Kumar Sangaiahd (2019), Classification of ransomware families with machine learning based on N-gram of opcodes, *Future Generation Computer Systems*, Volume 90, January 2019, Pages 211-221

[18] N.Andronio, S.Zanero and F.Maggi, "HelDroid: dissecting and detecting mobile Ransomware", *International Symposium on Recent Advances in Intrusion Detection*, pp. 382–404, 2015

[19] F.Mercaldo, V.Nardone, A.Santone andC.A.Visaggio, "Ransomware Steals Your Phone. Formal Methods Rescue It", *International Conference on Formal Techniques for Distributed Objects, Components, and Systems*, pp 212-221, 2016

[20] A.Kharraz, S.Arshad, C.Mulliner, W.K.Robertson, and E.Kirda, "UNVEIL: A Large-Scale, Automated Approach to Detecting Ransomware". *USENIX Security Symposium*, pp. 757–772, 2016

[21] S.Song, B.Kim, and S.Lee, "The effective ransomware prevention technique using process monitoring on android platform". *Mobile Information System*, pp. 3–11, 2016

[22] M.Egele, T.Scholte, E.Kirda, and C.Kruegel, "A survey on automated dynamic malware-analysis techniques and tools", *ACM Computing Survey*, vol. 44, no. 2, pp. 1-42, 2012

[23] T.Aragorn, C.YunChun, K.YiHsiang, and L.Tsungnan, "Deep learning for ransomware detection", *IEICE Technical Report; IEICE Tech. Rep*, vol. 116, no. 282, pp. 87-92, 2016

[24] A.K.Shaukat, and V.J.Ribeiro, "Ransomwall: A layered defense system against cryptographic ransomware attacks using machine learning", *IEEE 10th International Conference on Communication Systems \& Networks*, pp. 356–363, 2018

[25] S.Mirjalili, S.M.Mirjalili, A.Lewis, "Grey wolf optimizer", *Advance in Engineering Software*, vol. 69, pp.46–61, 2014

[26] X.Yang and SuashDeb, "Cuckoo Search via Lévy flights", *2009 World Congress on Nature & Biologically Inspired Computing (NaBIC)*, pp. 210-214, 2009

[27] J.Pedersen, D.Bastola, K.Dick, R.Gandhi, and W.Mahoney, "Blast your way through malware analysis assisted by bioinformatics tools", *International Conference on Security and Management*, 2012

[28] J.Xiao, J.Xu, Z.Chen, KaiZhang, and L.Pan, "A hybrid quantum chaotic swarm evolutionary algorithm for DNA encoding", *Computers and Mathematics with Applications*, Vol. 57, pp. 1949-1958, 2009

[29] S.Hao, J.Lu, P.Zhao, C.Zhang, S.C.H.Hoi and C.Miao, "Second-Order Online Active Learning and Its Applications", *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, no. 7, pp. 1338-1351, 2018

FIROZ KHANwas born in Kerala, India, in 1974. He received a BSc degree in Electronics from the Bharatiyaar University, Coimbatore, India, in 1991 and a Masters Degree in Information Technology from University of Southern Queensland, Australia, and another Master's Degree in Information Network and Computer Security (with Honors) from New York Institute of Technology, Abu Dhabi, UAE, in 2006 and 2016 respectively. He has completed his PhD in Computer Science from the British University in Dubai, Dubai, UAEresearching the field of

Ransomware detection. In 2001, he joined the Higher Colleges of Technology in Computer Information Science department as a Teaching Technician continued to become a Faculty member in 2005. He is currently holding the position of a Lecturer, with security and networking being his primary areas of teaching. His current research fields include computer security, machine learning, deep learning and computer networking.

CORNELIUS NCUBEis an Associate Professor of Computer Science at the British University in Dubai and an Honorary Fellow at the School of Informatics, Edinburgh University, UK. Previously, he was the Director of the Software Systems Research Centre, at the Faculty of Science & Technology, Bournemouth University, UK and the REF 2014 Unit of Assessment (UoA 11) Leader – Computing and Informatics submission. The Research Excellence Framework (REF) is the new system for assessing the quality of research in UK higher education institutions (HEIs). Cornelius holds a PhD in Computer Science (City, University of London, UK), an MSc in Software Engineering (City, University of London, UK) and a BSc (Hons) in Computer Science (Brunel University, London, UK). Cornelius's core research area is in Systems of Systems Engineering (SOSE) with a particular focus on Requirements Engineering and Securing Smart Cities. He has contributed to requirements engineering methods and software systems development processes. Special research recognition include winner of the prestigious IEEE Award: 'Most Influential Paper Award' (2008): The Award is for work on requirements engineering that had the most influence and impact on the theory or practice of Requirements Engineering in the last 10 years since its first publication in the IEEE Software Journal, 15(2), 46-56. The vote is undertaken annually by peers in the international requirements engineering community and reflects the value in work identified by the international community. Cornelius has been a lead researcher in various research projects; has served on the organisation and programme committees of several international conferences and workshops; guest-edited the November/December 2008 IEEE Software Journal on Opportunistic Software Systems Development, was Program Chair for the 7th IEEE International Conference on Composition-Based Software Systems (ICCBSS 2008) and has acted as a reviewer for various international journals and research project proposal. Professionally, he is a member of the International Council on Systems Engineering (INCOSE); the INCOSE Systems Security Engineering Working Group; the INCOSE Systems-of-Systems Engineering Working Group, the IEEE Computer Society and the Requirements Engineering Specialist Group of the British Computer Society (RESG).

LAKSHMANA KUMAR RAMASAMYis currently leading the technical team in Hindusthan College of Engineering and Technology, Coimbatore. Tamil Nadu. He is a global chapter Lead for MLCS [Machine Learning for Cyber Security]. Find him @ http://mlcsglobal.org/chapters-across-globe/. He is currently allied with company-specific training of Infosys Campus Connect, Oracle WDP and Palo Alto Networks. He himself involves in research and expertise in Fog computing and Blockchain technologies. In added to that, he is a lead guest editor for journals like Int. J. of Internet Technology and Secured Transactions (Indexed-Scopus (Elsevier) \& more) in Inderscience, International Journal of Gaming and Computer-Mediated Simulations (Indexed- Web of Science,Scopus (Elsevier),ESCI & more) in IGI Global, Ad Hoc & Sensor Wireless Networks (Indexed-SCI,Scopus &more), Open Computer Science (Indexed- Web of Science - Emerging Sources Citation Index, Scopus & more), IGI [International Journal of Knowledge and Systems Science (IJKSS)] (Indexed- Web of Science,Scopus (Elsevier),ESCI &more), Inderscience – International Journal of Data Science (IJDS), IGI – International Journal of Operations Research and Information Systems (IJORIS). He is acting as a vibrant reviewer for the journals like" IEEE Access (SCI indexed)", "Transactions on emerging telecommunications technologies (SCI indexed)", "International Journal of Web Services Research (IJWSR) (SCI indexed)"," International Journal of grid and utility computing" [Scopus], "International Journal of Computer-Aided Engineering and Technology "[Scopus], "World Review of Science, Technology and Sustainable Development" [Scopus], Journal of Information Systems Education" [Scopus] and many. He holds the certification in Data Science from John Hopkins University, United States. He also holds the Amazon Cloud Architect certification from Amazon Web Services.

SEIFEDINE KADRY received the bachelor’s degree in applied mathematics from Lebanese University, in 1999, the M.S. degree in computation from Reims University, France, and EPFL, Lausanne, in 2002, the Ph.D. degree from Blaise Pascal University, France, in 2007, and the HDR degree in engineering science from Rouen University, in 2017. He is currently working as an Associate Professor with Beirut Arab University,
Lebanon. His current research interests include education using technology, smart cities, system prognostics, stochastic systems, and probability and reliability analysis. He is a Fellow of IET, ACSIT, and ABET Program Evaluator.

YUNYOUNG NAM received the B.S., M.S., and Ph.D. degrees in computer engineering from Ajou University, South Korea, in 2001, 2003, and 2007, respectively. He was a Senior Researcher with the Center of Excellence in Ubiquitous System, Stony Brook University, Stony Brook, NY, USA, from 2007 to 2010, where he was a Postdoctoral Researcher, from 2009 to 2013. He was a Research Professor with Ajou University, from 2010 to 2011. He was a Postdoctoral Fellow with the Worcester Polytechnic Institute, Worcester, MA, USA, from 2013 to 2014. He has been the Director of the ICT Convergence Rehabilitation Engineering Research Center, Soonchunhyang University, since 2017, where he is currently an Assistant Professor with the Department of Computer Science and Engineering. His research interests include multimedia database, ubiquitous computing, image processing, pattern recognition, context-awareness, conflict resolution, wearable computing, intelligent video surveillance, cloud computing, biomedical signal processing, rehabilitation, and healthcare systems.