IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received October 15, 2021, accepted November 7, 2021, date of publication November 11, 2021,
date of current version November 24, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3127560

Efficient Classification of Enciphered SCADA Network Traffic in Smart Factory Using Decision Tree Algorithm

LOVE ALLEN CHIJIOKE AHAKONYE[1], (Member, IEEE), COSMAS IFEANYI NWAKANMA[2], (Member, IEEE), JAE-MIN LEE[2], (Member, IEEE), AND DONG-SEONG KIM[2], (Senior Member, IEEE)

[1]Networked System Laboratory, IT Convergence Engineering, Kumoh National Institute of Technology, Gumi 39177, South Korea
[2]Department of IT Convergence Engineering, Kumoh National Institute of Technology, Gumi 39177, South Korea

Corresponding author: Dong-Seong Kim (dskim@kumoh.ac.kr)

This work was supported in part by the Priority Research Centers Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (MEST) under Grant 2018R1A6A1A03024003; and in part by the Ministry of Science and Information and Communication Technology (MSIT), South Korea, under the Grand Information Technology Research Center support Program under Grant IITP-2021-2020-0-01612 supervised by the Institute for Information & Communications Technology Planning & Evaluation (IITP), South Korea.

ABSTRACT Vulnerability detection in Supervisory Control and Data Acquisition (SCADA) network of a Smart Factory (SF) is a high-priority research area in the cyber-security domain. Choosing an efficient Machine Learning (ML) algorithm for intrusion detection is a huge challenge. This study performed an investigative analysis into the classification ability of various ML models leveraging public cyber-security datasets to determine the best model. Based on the performance evaluation, all adaptions of Decision Tree (DT) and KNN in terms of accuracy, training time, MCE, and prediction speed are the most suitable ML for resolving security issues in the SCADA system.

INDEX TERMS Algorithms, artificial intelligence, machine learning, SCADA systems.

I. INTRODUCTION

Due to the present day smart factories' constant advancement in technology and its implementation, the need for advanced systems for the control of industrial processes either locally or at remote locations has become pertinent. The Supervisory Control and Data Acquisition (SCADA) system monitors, gather and process real-time data, thereby maintaining efficiency. It is required to ensure secured communication at every level of the system exactly connecting with systems like pumps, sensors, motors, valves, and more through the Human-Machine Interface (HMI) software. With the advent of cloud computing, smart factories, the Industrial Internet of Things (IIoT), and the addition of recent information technology models, protocols and usage such as web-based applications, the SCADA will continue to gain importance. However, its vulnerability is exploited due to a lack of efficient security and access authorization method [1]–[3].

In a 5G+ enabled smart factory, the Domain Name System (DNS) is the hub of communication and data transmission, assigning a chart between eligible host-names and the computer recognizable Internet Protocol (IP) addresses. Numerous firewalls do not scrutinize the recurrence and nature of DNS packets, which is a leeway for discreet data communication. Also, since the traffic packet goes through modified name-servers via a variety of hops past DNS iteration. Transmission is quite tough to detect and mitigate. This aids intruders take a certain measure of dominance over network devices and use them to plan attacks. Distinct perspectives have been used for the identification of irregular domain names and applying customary processes like IP boycott, domain boycott, and discarding suspicious DNS packets to attain DNS restriction [4]–[6]. Most security researchers critique DoH for making DNS channels difficult for detection and mitigation.

For safeguarding DNS traffic, the ideology of enciphering DNS over HTTPS, alternatively known as DoH for enhancing client authentication, access control security introduced [4]. Consider DoH sheathes the DNS packets in the DNS traffic, which is undetectable to the network framework between the DoH server and the malware. It constructively renders detection techniques based on investigating the DNS traffic

The associate editor coordinating the review of this manuscript and approving it for publication was Leandros Maglaras.

extinct for the firewalls. However, this technique did not eliminate DNS intrusion, considering the nature of $5G+$ connectivity.

Several alleviation methodologies exist, like firewall, Network-Based Intrusion Detection System (NB-IDS), SCADA hardware security and encryption methods [7]. NB-IDS has become a popular approach in reality and is widely used for security evaluation in networked systems (such as SCADA), thus the basis for this study. Studies exist that implemented various NB-IDS to tackle the vulnerability challenges in the SCADA. These NB-IDS techniques are Machine Learning-based (ML) due to their proficiency in the inherent process of identifying and categorizing attacks. Nevertheless, the premise for establishing the type of ML and its attributes creates a research challenge. There is always uncertainty on the need to select and confirm the best ML prospect.

Notwithstanding the attempts by researchers, there is no focus on a decisive choice of the apt ML algorithm for SCADA vulnerability detection. Hence, developers face uncertainty and predicament in arriving at the algorithm of choice. Therefore, some analytical questions will be; what strategy and measures impressed the ML prospects? What are the reasons for the poor performances of ML techniques? Moreover, in the case of similar achievement between two ML techniques, what arouses the decision for the preferred ML prospect? This study aims to confront the above.

Prompted by the above problems, the aim of this study is to accomplish the following:

1) To carry out assessment of discrete ML algorithms effective for SCADA network attack detection. Presenting arithmetical intuition into the rationale for the ML algorithm and its achievement. It is pertinent to researchers as it aids developers to focus on the choicest and most suitable ML approach for their studies.
2) Demonstrating the determination and application of the most suitable ML algorithms effective for SCADA network attack detection.
3) Conclusively arrived at the option of an excellent performed and light-weighted ML algorithm based on computational complexity and accuracy in model training time, system constraints considering controls in the processing ability of industrial equipment and operations of a Smart factory.

The rest of the study is organized thus: the background study is followed by Section II, related works which give insight into current studies on ML IDS classification algorithms. Then Section III describes the concept of algorithm and methodology of the study. In Section IV, performance evaluation and conscientious analysis of various ML prospects simulated were presented. The rest of the study was concluded in Section V with a presentation of a prospective NIDS in the SCADA network system.

II. RELATED WORKS

Authors in [8]–[10] presented studies on current attacks on cyber-security and mitigation approaches using machine and deep learning (ML/DL). This studies highlighted the benefits of the use of ML and DL in IDS. Collective researchers in [11], [12] examined the recent advancements in cyber-security risk evaluation in application to SCADA systems utilizing entrenched investigation procedures. The study contains a variety of security and vulnerability linked to research on SCADA. In [13], authors provided an extensive review on approaches that can be applied for vulnerability detection in the system and ascertaining the extent of safeguarding against possible attacks. Examples of such approaches presented in this work are simulating SCADA attacks, testbeds simulation frameworks, etc.

The internet connectivity proffers numerous functional services such as remote interconnection and flexibility; contrarily, it endangers systems such as SCADA by making them prone to the vulnerability of global attacks on cyber-security. Therefore, several studies on IDS algorithms for SCADA vulnerability were accounted for and reported in the literature. Ensemble learning (EL) approaches use a blend of distinct classifiers to make predictions. This yield enhanced performance for various attack types and protocols used in IoT networks [14]–[16] Though this system delivered superior results, the approach can be complex with a lack of computational speed. Reference [17] presented an autoencoder-based IDS targeted at the Distributed Network Protocol 3 (DNP3) of a SCADA system. The proposed model had better performance when compared with other IDS solutions. However, this achievement cannot be generalized due to restricted usage in a small dataset and specific DNP3 operation environments.

In recent times, the Convolutional Neural Network (CNN) framework in DL has made a tremendous impact on computer vision. It is associated with the two most important features; hierarchical feature representations and learning long-term dependencies in large-scale sequence data. The study by [18] presented an approach for detecting attacks using Long Short-Term Memory (LSTM) and CNN, evaluating the popular conventional NSL-KDD dataset. Though the model had an excellent performance, the classification speed needs improvement. In the same vein, [4] proposed an approach that attempts to recognize and classify DNS over Hypertext traffic in two layers using classifiers. The authors claimed that the main superiority of the study is the ability to avidly detects and classifies DoH traffic using a small amount of input data. Hence, not a robust model therefore not suitable for Smart factory operations. Another study by [19] presented a hybridization method with universal optimization approach for detecting DDoS attacks in IoT. This approach evaluated the early version of the CICIDS2017 dataset. Though the model seemed efficient, lacked computational speed and is not robust enough for a Smart factory, authors hope to try it on distributed IDS.

Sundry ML algorithms have emanated as series of studies addressed the development of IDS. It is to determine and deploy the best viable and efficient algorithms. Reference [20] in a study presented a flow-based intrusion detection research for a SCADA system using deep Artificial Neural Network (ANN). The proposed model evaluated attacks online and offline. The approach had an excellent performance. However, it requires an extension to multiple attacks. In [7], the authors used the Decision Tree (DT), Random Forest (RF), Naive Bayes (NB), Logistic Regression (LR) and K-Nearest Neighbour (KNN) classifiers on a water storage facility testbed generated dataset. From their evaluation, utilized classifiers had a good performance in accuracy and false alarm rates for offline and online phases with an unbalanced dataset. However, the study did not specify the training time of the model. It is important to note that for critical industrial systems, accuracy alone is not the ideal measure to assess the capability of a model; other metrics such as false alarm rate, training time are needed for comparison.

Besides, [21] in a study, attempted to compare the performance of NB, KNN, ANN techniques. The individual classifiers performed poorly. Hence, the ensemble technique was employed to enhance performance. In another study [22], the authors implemented an ensemble approach to boost performance. The trial results show that the proposed ensemble framework copes well in precision, accuracy, recall and detection rate. However, this fusion requires sparse computational cost as one of the factors for time-critical operations.

III. SYSTEM MODEL

A. UNDERSTANDING ALGORITHMS AND GROWTH RATE FUNCTIONS

An algorithm is a sequence of computational steps that transforms input to output, applied in situations that require human ingenuity. It provides efficient solutions for emerging trends in IDS. Determine the best solution in addition to stabilizing memory management and running time in a computational environment. In choosing an algorithm, efficiency and preciseness are apt. It can also be affected by the implementation of hardware and software. However, having a solid base of algorithmic knowledge and its techniques is vital.

In implementing algorithms, a good understanding of the relationship among features of variables in a dataset is crucial. This guides in choice of solution based on the algorithm growth rate. Analysis of an algorithm helps to determine the behavioural pattern as input size is affected by variation. This change in the behaviour of an algorithm is known as the asymptotic growth rate. Stated below are the equations for algorithm growth functions:

$$Linear : Q = aM + x, \tag{1}$$
$$Logarithm : Q = alogM + x, \tag{2}$$

TABLE 1. Algorithm growth rate functions *F*(n).

Name	Computation Time	Performance
Linear	O(m)	Good
Logarithm	O(logm)	Very Good
Cubic	$O(m^3)$	Poor
Quadratic	$O(m^2)$	Acceptable
Exponential	$O(2^m)$	Bad
M-Logarithm	O(m.logm)	Fair

TABLE 2. Algorithm function descriptions.

Name	Functions
Linear	Search, delete and insert operations
Logarithm	Utilized for binary operations with logarithmic behaviour
Cubic	Traversing 3 dimensional structure
Quadratic	Nested loops scenario (m x m matrix)
Exponential	Utilized in most applications
M-Logarithm	Divide and conquer

$$Cubic : Q = aM^3 + x, \tag{3}$$
$$Quadratic : Q = aM^2 + x, \tag{4}$$
$$Exponential : Q = a2^m + x, \tag{5}$$
$$M - Logarithm : Q = a\,(x)\,log + x, \tag{6}$$

where a and x are constant, M is input size. This implies that an increase in the size of M, influences the value of a and x. Consider the asymptotic growth rate based on the Big (O) notation, see Table 1 for functions of algorithm growth rate. Table 2 shows the description of the actions performed by the various algorithms.

With the rise of intricacy, SCADA has become so relevant that it requires safeguarding. IDS in SCADA helps recognize real-time observable frontiers, filter DNS traffic between established and un-established name servers with advance or discard guidelines. IDS in SCADA has been handled by different studies as presented in Section I. This work analyzed algorithmic functions and growth rate in addition to a comparison of various ML algorithms. The ML algorithms were developed to monitor SCADA network traffic and identifying the abnormal nature in the traffic to address security issues in the $5G+$ enabled smart factory.

The proposed architecture consists of 3 main phases; training, testing, and model selection phase, see Fig 1. During the training and testing phase, the various datasets used in this study is split into training and testing sets and imported into the ML algorithms after implementing a five (5) fold cross-validation. The testing set was used to validate the performance of the training set. The Performance Metrics (PM) used in evaluating the models are Accuracy of the model, Receiver Operating Characteristics (ROC), Confusion Matrix (CM), training time, and the model's Mis-Classification Error (MCE). These PMs guided the selection of the best model.

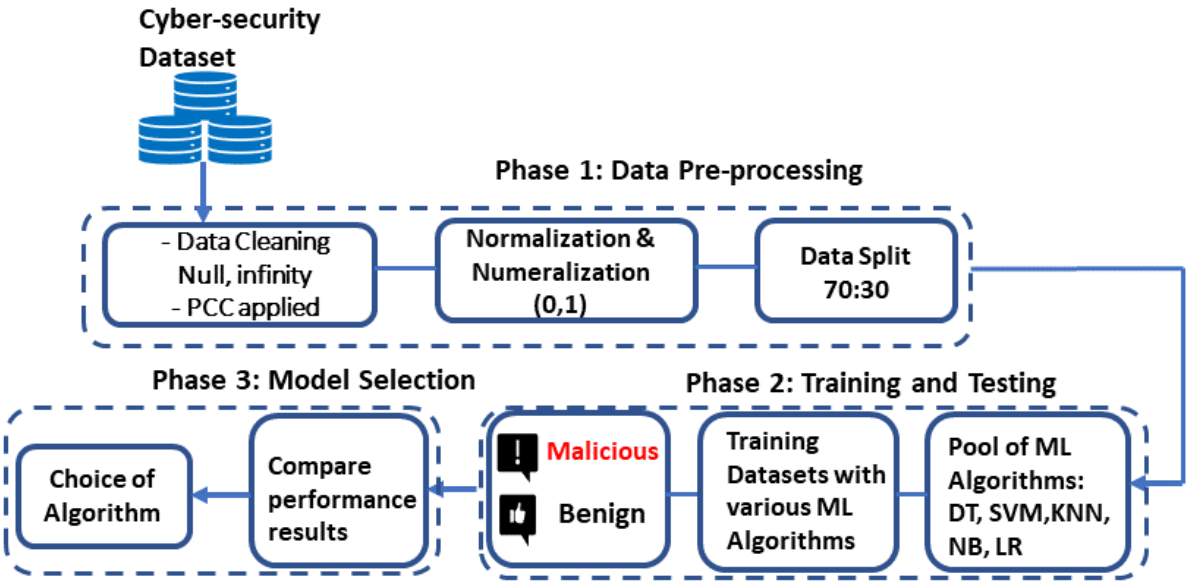


FIGURE 1. Model design displaying the phases of the ML technique evaluation for SF SCADA attack classification.

B. CASE STUDY/DATASET DESCRIPTION

This study applied three (3) dataset types as case studies to evaluate the different ML algorithms' efficiency and performance. It enables the validation and determination of the most suitable solution in the prevailing emerging vulnerability situations. The data for each case study was split into 70%:30% for training and test validation. For even data representation, data-balancing was done using the SMOTE technique. Below is the description of the evaluated datasets.

1) CASE STUDY 1 (CIRA-CIC-DoHBrw-2020 DATASET)

The preprocessed cyber-security intrusion dataset (CIRA-CIC-DoHBrw-2020) is accessible at [23]. The dataset was generated by the use of web browsing activities for the benign-DoH. While DNS channeling mechanisms were used for generating malicious-DoH traffics, [4]. The cyber-security data traffic contains a total of 226406 observations, 28 predictors and two (2) responses represented as benign and malicious scenarios.

2) CASE STUDY 2 AND 3 (NSL-KDD DATASET)

NSL-KDD dataset is an offshoot after analyzing the KDD cup'99 dataset, prior to this, KDD is a conventional dataset. The challenge of the KDD cup'99 dataset is the presence of redundant data, which is biased towards repeated data. These issues were addressed hence, the NSL-KDD dataset was proposed [18]. Afterwards, it is being used as a standard dataset. The dataset is divided into KDDTest+ and KDDTrain+ comprising of a total of 22543 observations and 41 predictors and 2 responses represented as anomalous and normal scenarios.

In the training phase of the model, the dataset is fed into the model followed by data pre-processing, which is explained in the next in the subsequent subsection as follows:

C. DATA PRE-PROCESSING

This stage starts with data cleaning, replacing fields with infinity (∞) and nulls in the column with the mean value of that column. It ensures that only meaningful values get passed into the model. Subsequently, since the dataset contains correlated features as seen in the correlation matrix represented in Fig. 2, the Pearson's Correlation Coefficient (PCC) was performed on the dataset. This technique was necessary as it ensures the reduction of over-fitting. PCC was implemented for consecutive variables, with a correlation score between -1 and 1 as represented in equation 7, with the selection of variables with a high correlation value at a threshold of $+/-1$. This aids in ensuring that only reliably significant features are selected, thereby enhancing the model performance. Fig. 3 depicts the selection of the correlated variables using a threshold of $+/-1$.

$$X = \frac{\sum \left(a_i - \hat{a}\right)\left(b_i - \hat{b}\right)}{\sqrt{\sum \left(a_i - \hat{a}\right)^2 \left(b_i - \hat{b}\right)^2}}, \tag{7}$$

where X depicts the Pearson Correlation Coefficient, a_i, are content of the variables in the dataset, $\hat{a}$, represents the mean values of the a variables, b_i is the variables of the sample and $\hat{b}$ shows the mean values in the b variables.

This study had no application of feature selection as a result of no presence of irrelevant features in the dataset. All existing features contributed to the model's decision-making. The feature selection technique is used for the

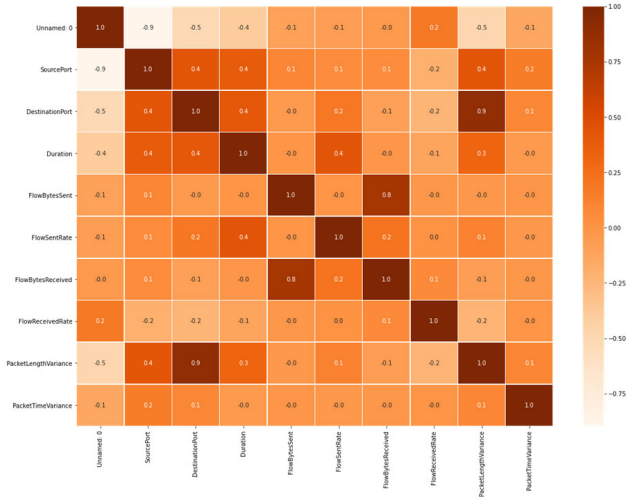


FIGURE 2. Correlation matrix showing highly correlated features.

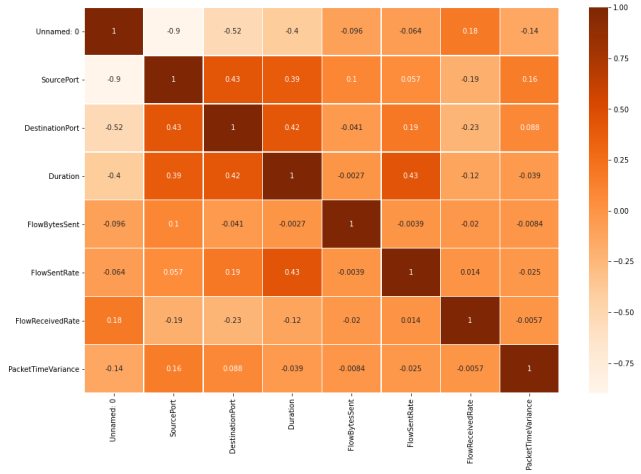


FIGURE 3. Result of PCC validating the feature importance selection.

removal of redundant features by selecting the most promising. This is also an important issue in model design, developers need to understand the nature and role of data features to be able to design efficient model for time-critical systems. It will aid the avoidance of unnecessary humongous models.

D. DESCRIPTION OF MACHINE LEARNING PROSPECTS

The experiment was carried out on MATLAB R2019b, Table 3 contains features of all analyzed ML prospects such as number of responses, total number of observations and predictors of samples of each case study, evaluated models and features, K-validation and result presentation. The following ML algorithms were examined:

1) Trees: In this model, prediction of the value of an objective attribute is confirmed. A depiction of a tree is applied to ascertain basic decision precepts deduced from data characteristics. To achieve this, the internal node of the Tree is represented by the leaf node

TABLE 3. Specifications of models architecture.

Features of Model Parameters	
Models	*Parameters*
Observations	Case Study 1 = 226406 samples each Case Study 2 and 3 = 22543 samples
Predictors	Case Study 1 = 28 features Case Study 2 and 3 = 41 features each
Responses	Case Study 1, 2 and 3 = 2 responses each
Result	Confusion Matrix, ROC
Trees	Max Splits = 100, Split Criterion = Gini's Diversity Index Preset = FT, MT, CT, OPT
KNN	Neighbours = 1 & 10, Distance Weight = Squared inverse, Equal Distance Metric = Euclidean Preset = Weighted KNN, Fine KNN
ELM	Max Splits = 20, Method = AdaBoost Learning Rate = 0.1 Preset = Boosted, Bagged, RSUBoosted Learning Type = Decision Tree
SVM	Kernel Function = Linear, Gaussian, Kernel Scale = 0.00101952, 1, Automatic Box Constraint Level = 1.75813, 1, 26 Multi class Method = One Vs One, One Vs All Preset = CGSVM, MGSVM, FGSVM
K-Validation	5 Fold

comparable to a class label and characteristics. Assess unique fixed values each for n range $A_1 - A_n$ as indicated in equation 8, then $Q_i(B) = 1$ when B in Ri This enables the prediction of a class A based on a feature B by a model.

$$\hat{A}(B) = \sum_{i=0}^{n} A_i * Q_i(B), \tag{8}$$

2) Support Vector Machines (SVM): This classifier builds a class of resolution confines which aids to classify data values. A division is obtained by the conclusion confines with a huge interval to the closest training data position of any class. Hence, the generalization error of the classifier is determined the extent of the margin (either higher or lower). Given a training set,

$$a_i \in Q_p, \quad for\ i = 1, \ldots n$$

in two sets, and a variable $y \in (1, -1)^n$. The aim is to ascertain

$$\alpha \in Q^p \ \ and \ \ b \in Q$$

such that the prediction given by $\alpha^T \phi(a) + b)$ is accurate for many samples. SVM is represented as shown in equation 9.

$$\min_{\theta, b, \varphi} \frac{1}{2}\theta^T\theta + D\sum_{i=0}^{z}\varphi, \tag{9}$$

subject to

$$y_i(\theta^T\phi(a_i + b) \geq 1 - \varphi_i, \quad \varphi_i = 1, \ldots, n$$

3) Ensemble Learning (EL) Algorithms: The objective this algorithm as depicted in equation 10 is the conjugation of constant evaluators into a major one with the goal of enhancing its reliability over an individual evaluator.

$$\varrho = \bigsqcup(\rho, \beta, \in, \sigma), \tag{10}$$

where ρ is the dataset array, β shows the variable feedback. $\in$ is the method of combination, σ marks the diverse hyper-parameter tuning and types of learners while ϱ represents the EL method. Then, $\bigsqcup$ outlines the correlation between the attributes.

4) Discriminant Models (DM): A DM learns to model euclidean spaces among classes. It compares prototype attributes to class labels, and detects the coinciding viability of such prototypes and their associating classes. This model is obtained from plain anticipated model of a class state data allotment; $Q\ (H|y = t)$ for each class t. Conclusions for individual training specimen, $t \in S^p$ can be obtained by using equation 11;

$$Q\,(y = t) = \frac{Q\,(h \setminus y = t)\,Q\,(y = t)}{\sum_Q (h \setminus y = r)\,.Q\,(y = r)}, \tag{11}$$

5) K-Nearest Neighbour (KNN): In this model, the goal is to learn a pre-established number of training samples with close proximity in distance to the new point using the Euclidean metrics in most cases as shown in equation 12, and predict the tag from it.

$$(x, x!) = \sqrt{(x1, -x1!)^2 + \ldots + (xm, -xm!)^2}, \tag{12}$$

6) Naive Bayes (NB): The model is simply a probability directory that is refreshed subject to training data. Prediction is made based on the observation of the class probabilities in the probability directory based on the values of the attributes. Given a feature vector $Y = (y_1, y_2, \ldots, y_n)$ and a class variable Ck, equation 13 states that:

$$Q\,(c|Y) = Q\,(y_1|c) * Q\,(y_2|c) * \ldots * Q\,(y_m|c) * Q\,(c), \tag{13}$$

for $k = 1, 2, \ldots, K$.

7) Logistic Regression (LR): In this model, there is a direct association, between the input and their commensurate output. The coefficient of LR model is always evaluated from the training and dissemination of the data. See equation 14.

$$c = \Theta_0 + \Theta_1 a_1 + \Theta_1 a_1 + \ldots \Theta_m a_m, \tag{14}$$

IV. PERFORMANCE EVALUATION

Accuracy appears to be the most accessible metric when choosing an algorithm for an ML task. Nevertheless, accuracy alone is not enough to help in selecting the best algorithm. The model needs to meet other conditions, such as data relationship, training and prediction time, interpret-ability, data format and other performance metrics. A combination of a broad scope of these factors aids in making a more confident decision. This study makes use of the combined advantage of the model training time, mis-classification error (MCE), prediction speed and accuracy as the basis for arriving at the choice of a light-weight ML algorithm for attack detection in the SCADA network traffic.

A. EVALUATION OF THE DECISION TREES TECHNIQUE

In the class of DT, Fine, Coarse, Medium, and Optimizable were analyzed for all the datasets. The Fine, Medium, Coarse, and Optimizable Trees had an accuracy of 98.1%, 97.0%, 95.0%, and 99.1% for the CIRA-CIC-DoHBrw-2020. NSL-KDDTest had an accuracy of 97.8%, 96.8%, 92.9% and 98% and NSL-KDDTrain with 99.5%, 98.5%, 96.2% and 99.7% respectively. It shows that considering Trees as a scheme assures a minimum of 95.0% accuracy irrespective of type. Also, since the difference in accuracy between Optimizable and Fine Tree is minute, it is obvious that the choice of Fine Tree is the best preference without the necessity for optimization.

B. EVALUATION OF ENSEMBLE LEARNING METHOD (ELM)

A comparison of the performance of the EL algorithms and comparing it to default schemes was investigated. This study examined the EL of several machine learning. The Ensemble Bagged Trees recorded 99.4% accuracy at a training time of 85.488s, confirming the superior performance of the Trees. Though, EL Subspace KNN had 99.2% while the least was ensemble subspace discriminant with a low accuracy of 92.8% in the CIRA-CIC-DoHBrw-2020 dataset. However, for NSL-KDDTest+, only the Bagged, Boosted and RSUBoosted Trees performed. Recording accuracy of 97.9%, 97.1% and 96.8%, respectively. It further confirms the suitability of the Trees algorithm.

C. EVALUATION OF SUPPORT VECTOR MACHINES (SVMs) TECHNIQUE

This study examined all adaptions of SVM. The evaluation shows that for all the datasets, Fine Gaussian, Medium and Coarse Gaussian SVMs recorded the best accuracy in contrast to Linear, Quadratic and Cubic SVM, which recorded the lowest accuracy in all three (3) datasets. This technique is best for modelling decision boundaries which are non-linear [24].

D. EVALUATION OF LOGISTIC REGRESSION (LR) TECHNIQUE

The LR recorded accuracy of 96.3%, 93.6% and 91.5% respectively for NSL-KDDTrain+, CIRA-CIC-DoHBrw-2020 and NSL-KDDTest+ datasets. It is not acceptable considering the significance and intensity of accuracy in forestalling SCADA network attacks. Besides, LR is most fitting in a probabilistic classification relationship rather than linear [25].

TABLE 4. Comparative analysis of 3 dataset used highlighting 3 best performed ML techniques based on accuracy, time, MCE and prediction speed.

Performance Metrics	CIRA-CIC-DoHBrw-2020	NSL-KDD (Tes+)	NSL-KDD (Train+)
Accuracy (%)	Weighted KNN = 99.2	Fine Tree = 97.7	Fine Tree = 99.6
	Fine KNN = 99.2	Medium Tree = 96.8	Medium Tree = 98.5
	Fine Tree = 98.1	Coarse Tree = 92.9	Coarse Tree = 96.2
Training Time (s)	Weighted KNN = 8.8344	Fine Tree = 4.1465	Fine Tree = 8.1022
	Fine KNN = 13.583	Medium Tree = 1.7494	Medium Tree =4.2779
	Fine Tree = 4.5778	Coarse Tree = 1.2456	Coarse Tree =4.2446
MCE (#)	Weighted KNN = 2134	Fine Tree = 525	Fine Tree = 560
	Fine KNN = 2192	Medium Tree = 722	Medium Tree = 1896
	Fine Tree = 5096	Coarse Tree = 1609	Coarse Tree = 4825
Prediction Speed (obs/sec)	Weighted KNN = 250000	Fine Tree = 150000	Fine Tree = 370000
	Fine KNN = 440000	Medium Tree = 230000	Medium Tree = 450000
	Fine Tree = 1100000	Coarse Tree = 230000	Coarse Tree = 430000

TABLE 5. Comparative analysis of 3 dataset used highlighting 3 least performed ML techniques based on accuracy, time, MCE and prediction speed.

Performance Metrics	CIRA-CIC-DoHBrw-2020	NSL-KDD (Tes+)	NSL-KDD (Train+)
Accuracy (%)	Cubic SVM = 44.6	Gaussian Naive Bayes = 86.7	Gaussian Naive Bayes = 77.3
	Linear SVM = 36	Quadratic SVM = 68.2	Quadratic SVM = 68.2
	Quadratic SVM = 10.6	Cubic SVM = 43.8	Cubic SVM = 43.8
Training Time (s)	Cubic SVM = 10046	Gaussian Naive Bayes = 2.2045	Gaussian Naive Bayes = 11.019
	Linear SVM = 7252.9	Quadratic SVM = 719.51	Quadratic SVM = 719.51
	Quadratic SVM = 5742.1	Cubic SVM = 806.3	Cubic SVM = 806.3
MCE (#)	Cubic SVM = 149331	Gaussian Naive Bayes = 3008	Gaussian Naive Bayes = 28609
	Linear SVM = 172670	Quadratic SVM = 7175	Quadratic SVM = 7175
	Quadratic SVM = 241081	Cubic SVM = 12661	Cubic SVM = 12661
Prediction Speed (obs/sec)	Cubic SVM = 250000	Gaussian Naive Bayes = 210000	Gaussian Naive Bayes = 42000
	Linear SVM = 3000	Quadratic SVM = 33000	Quadratic SVM = 33000
	Quadratic SVM = 3300000	Cubic SVM = 16000	Cubic SVM = 16000

E. EVALUATION OF DISCRIMINANT ANALYSIS TECHNIQUE

The investigation reveals that the optimizable and quadratic discriminant had 92.9% accuracy. While linear discriminant options and 92.8% respectively for CIRA-CIC-DoHBrw-2020. However, the algorithm failed in NSL-KDDTrain+ and NSL-Test+. Consequently, the result depicts the unsuitability of discriminant ML for attack detection and classification. It is in the ensemble subspace discriminant approach and the non-applicability of the algorithm in NSL-KDDTest and NSL-KDDTrain datasets.

F. EVALUATION OF K-NEAREST NEIGHBOURS (KNN) TECHNIQUE

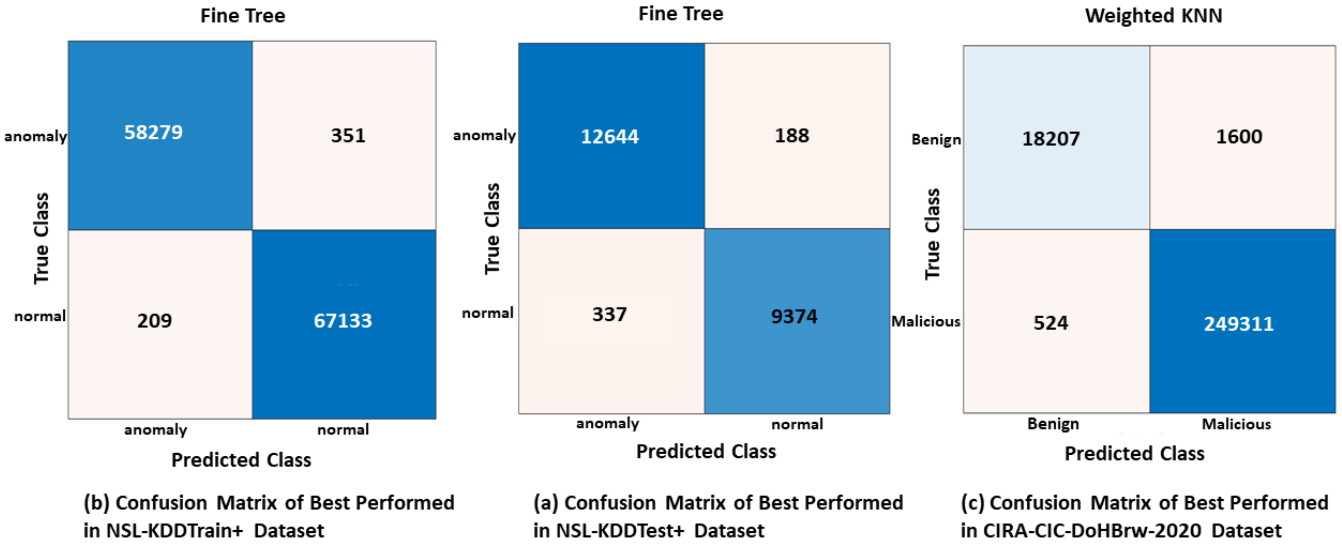
In the MATLAB toolbox, the KNN technique comprised Fine, Medium, Cosine, Coarse, Cubic, Weighted and Optimizable. All the KNN algorithms had above 99% accuracy except the Cosine and Coarse KNN with 98.8% and 98.4% accuracy respectively in the CIRA-CIC-DoHBrw-2020 dataset. Though with high interpretability through feature importance, it is most suitable with linearly related data. However, it was observed that this algorithm is not fit and inapplicable to NSL-KDDTrain+ and NSL-KDDTest+ datasets. It is, therefore, evident that the KNN algorithm is sensitive to features of data relationships.

G. EVALUATION OF NAIVE BAYES (NB) TECHNIQUE

The Kernel Naive Bayes (KNB) recorded a not too encouraging accuracy in the three (3) datasets except for the optimizable Naive Bayes (OGNB) with an accuracy of 96% and 96.5% across the datasets, respectively. However, this shows that the NB is not fit for countering attacks in the SCADA network.

H. REVIEW OF THE ANALYSIS OF PERFORMANCE EVALUATION

In selecting an ML algorithm, knowing how to make the right choice that is most suitable for the specific problem is imperative. A comprehensive understanding of the relationship existing amongst the data features is critical in decision making. Considering data relationship, training and prediction time, interpret-ability, and data format, etc. a total of twenty-nine (29) models were simulated. The result of the evaluation for the best and least performed techniques leveraging on three (3) cyber-security datasets is as shown in Tables 4 and 5 in terms of accuracy, training time, MCE and prediction speed. A comparative analysis of the three (3) cyber-security datasets was carried out for further validation of the proposed choice of models for SCADA IDS, see Fig. 4 for the confusion matrix of the best-performed algorithm in the respective datasets and Fig. 5 shows the Receiver

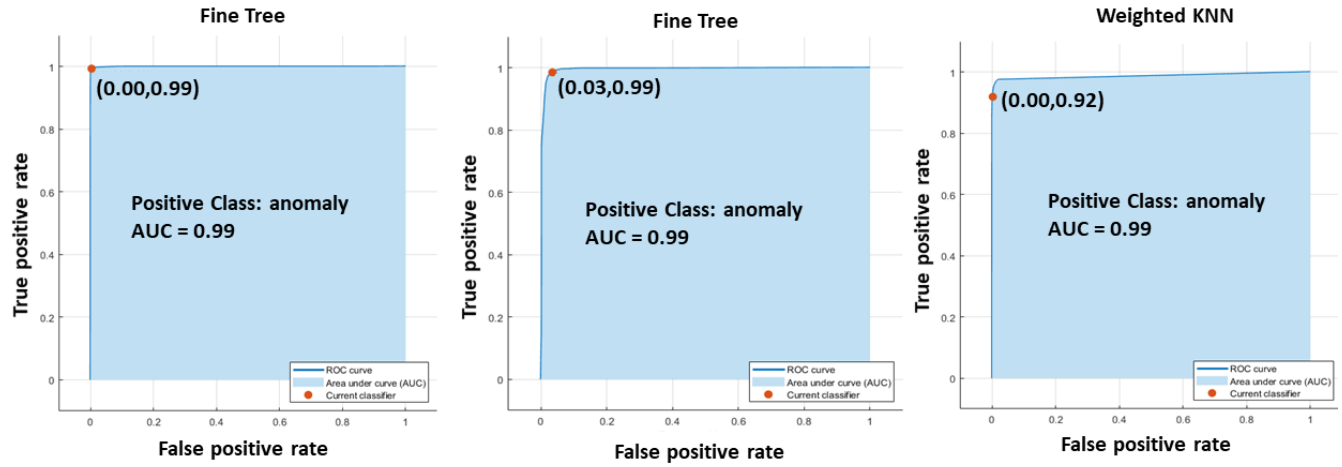


(b) Confusion Matrix of Best Performed in NSL-KDDTrain+ Dataset

(a) Confusion Matrix of Best Performed in NSL-KDDTest+ Dataset

(c) Confusion Matrix of Best Performed in CIRA-CIC-DoHBrw-2020 Dataset

FIGURE 4. Confusion matrix showing the best performed algorithms in the three (3) compared cyber-security datasets.



(a) ROC of Best Performed in NSL-KDDTrain+ Dataset

(a) ROC of Best Performed in NSL-KDDTest+ Dataset

(c) ROC of Best Performed in CIRA-CIC-DoHBrw-2020 Dataset

FIGURE 5. ROC showing the best performed algorithms in the three (3) compared cyber-security datasets.

Operating Characteristic curve (ROC) of the best-performed algorithm across the three (3) datasets based on the MCE. An ROC curve is a graph displaying the performance of the top three classification models. The curve plots two parameters namely True Positive Rate (TPR) and False Positive Rate (FPR) of a model.

V. CONCLUSION

This study presents an evaluation of various ML techniques for the Smart Factory SCADA network in terms of accuracy, training time, MCE and prediction speed using a trade-off in time and accuracy. It is imperative to note that for the time-critical target domain, accuracy alone does not suffice as a measure to determine the model's performance. Also, following the non-existence of redundant data features, the model did not apply feature selection.

From the comparative analysis of three (3) state-of-art cyber-security datasets, the evaluation result shows that all adaptions of the Decision Trees (DT) and K-Nearest Neighbours (KNN) are the most suitable for vulnerability detection. All classes of DT and KNN presents a combined advantage of good accuracy, least training time, MCE and observation per second. Thus, the DT and KNN algorithms had the best performance in detecting and classifying attacks in the Smart Factory SCADA network. It is shown by the rapid classification and high predictive capacity of the model with superior accuracy, time-efficiency, ease of interpretation and better stability which, is occasioned by adequately

mapping non-linear relationships effectively. The Gaussian Naive Bayes, Cubic, Linear and Quadratic SVM performed poorly in training time, accuracy, MCE and observation per second during evaluation. The poor performance of this class of SVM is due to their non-linear inclination to classification, utilizing the kernel trick hence favour pattern analysis.

In contrast to compared related works, where EL models were employed to improve performance without determining suitable ML algorithms. This study aimed at deducing models fit for classification problems and insight into arithmetical rationale to the performance of the model so as to guide researchers and developers in choice of models.

REFERENCES

[1] D.-S. Kim and H. Tran-Dang, "Industrial sensors and controls in communication networks," in *Computer Communications and Networks*. Cham, Switzerland: Springer, 2019.

[2] C. I. Nwakanma, F. B. Islam, M. P. Maharani, J.-M. Lee, and D.-S. Kim, "Detection and classification of human activity for emergency response in smart factory shop floor," *Appl. Sci.*, vol. 11, no. 8, p. 3662, Apr. 2021. [Online]. Available: https://www.mdpi.com/2076-3417/11/8/3662

[3] G. C. Amaizu, C. I. Nwakanma, S. Bhardwaj, J. M. Lee, and D. S. Kim, "Composite and efficient DDoS attack detection framework for B5G networks," *Comput. Netw.*, vol. 188, Apr. 2021, Art. no. 107871. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1389128621000438

[4] M. MontazeriShatoori, L. Davidson, G. Kaur, and A. H. Lashkari, "Detection of DoH tunnels using time-series classification of encrypted traffic," in *Proc. IEEE Int. Conf. Dependable, Autonomic Secure Comput., Int. Conf. Pervasive Intell. Comput., Int. Conf. Cloud Big Data Comput., Int. Conf. Cyber Sci. Technol. Congr. (DASC/PiCom/CBDCom/CyberSciTech)*, Calgary, AB, Canada, Aug. 2020, pp. 63–70.

[5] A. Nadler, A. Aminov, and A. Shabtai, "Detection of malicious and low throughput data exfiltration over the DNS protocol," *Comput. Secur.*, vol. 80, pp. 36–53, Jan. 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167404818304000

[6] C. Patsakis, F. Casino, and V. Katos, "Encrypted and covert DNS queries for botnets: Challenges and countermeasures," *Comput. Secur.*, vol. 88, Jan. 2020, Art. no. 101614. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S016740481831321X

[7] M. Teixeira, T. Salman, M. Zolanvari, R. Jain, N. Meskin, and M. Samaka, "SCADA system testbed for cybersecurity research using machine learning approach," *Future Internet*, vol. 10, no. 8, p. 76, Aug. 2018, doi: 10.3390/fi10080076.

[8] Z. Ahmad, A. S. Khan, C. W. Shiang, J. Abdullah, and F. Ahmad, "Network intrusion detection system: A systematic study of machine learning and deep learning approaches," *Trans. Emerg. Telecommun. Technol.*, vol. 32, no. 1, p. e4150, Jan. 2021, doi: 10.1002/ett.4150.

[9] A. S. Khan, Z. Ahmad, J. Abdullah, and F. Ahmad, "A spectrogram image-based network anomaly detection system using deep convolutional neural network," *IEEE Access*, vol. 9, pp. 87079–87093, 2021.

[10] N. Mishra and S. Pandya, "Internet of Things applications, security challenges, attacks, intrusion detection, and future visions: A systematic review," *IEEE Access*, vol. 9, pp. 59353–59377, 2021.

[11] Y. Cherdantseva, P. Burnap, A. Blyth, P. Eden, K. Jones, H. Soulsby, and K. Stoddart, "A review of cyber security risk assessment methods for SCADA systems," *Comput. Secur.*, vol. 56, pp. 1–27, Feb. 2016. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167404815001388

[12] F. A. Alhaidari and E. M. AL-Dahasi, "New approach to determine DDoS attack patterns on SCADA system using machine learning," in *Proc. Int. Conf. Comput. Inf. Sci. (ICCIS)*, Apr. 2019, pp. 1–6.

[13] S. Nazir, S. Patel, and D. Patel, "Assessing and augmenting SCADA cyber security: A survey of techniques," *Comput. Secur.*, vol. 70, pp. 436–454, Sep. 2017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167404817301293

[14] C. Yue, L. Wang, D. Wang, R. Duo, and X. Nie, "An ensemble intrusion detection method for train Ethernet consist network based on CNN and RNN," *IEEE Access*, vol. 9, pp. 59527–59539, 2021.

[15] N. Moustafa, B. Turnbull, and K.-K.-R. Choo, "An ensemble intrusion detection technique based on proposed statistical flow features for protecting network traffic of Internet of Things," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4815–4830, Jun. 2019.

[16] A. Abbasi, A. R. Javed, C. Chakraborty, J. Nebhen, W. Zehra, and Z. Jalil, "ElStream: An ensemble learning approach for concept drift detection in dynamic social big data stream learning," *IEEE Access*, vol. 9, pp. 66408–66419, 2021.

[17] M. Altaha, J.-M. Lee, A. Muhammad, and S. Hong, "An autoencoder-based network intrusion detection system for the SCADA system," *J. Commun.*, vol. 16, pp. 210–216, Jun. 2021.

[18] L. Karanam, K. K. Pattanaik, and R. Aldmour, "Intrusion detection mechanism for large scale networks using CNN-LSTM," in *Proc. 13rd Int. Conf. Develop. eSyst. Eng. (DeSE)*, Dec. 2020, pp. 323–328.

[19] M. Roopak, G. Y. Tian, and J. Chambers, "An intrusion detection system against DDoS attacks in IoT networks," in *Proc. 10th Annu. Comput. Commun. Workshop Conf. (CCWC)*, Jun. 2020, pp. 0562–0567.

[20] M. A. Teixeira, M. Zolanvari, K. M. Khan, R. Jain, and N. Meskin, "Flow-based intrusion detection algorithm for supervisory control and data acquisition systems: A real-time approach," *IET Cyber-Phys. Syst., Theory Appl.*, vol. 6, no. 3, pp. 178–191, Sep. 2021, doi: 10.1049/cps2.12016.

[21] D. Upadhyay, J. Manero, M. Zaman, and S. Sampalli, "Intrusion detection in SCADA based power grids: Recursive feature elimination model with majority vote ensemble algorithm," *IEEE Trans. Netw. Sci. Eng.*, vol. 8, no. 3, pp. 2559–2574, Jul. 2021.

[22] A. H. Mirza, "Computer network intrusion detection using various classifiers and ensemble learning," in *Proc. 26th Signal Process. Commun. Appl. Conf. (SIU)*, May 2018, pp. 1–4.

[23] (2020). *CIRA-CIC-DoHBrw-2020 Dataset*. [Online]. Available: https://www.unb.ca/cic/datasets/dohbrw-2020.html

[24] R. Shirkin, *Artificial Intelligence: The Complete Beginners' Guide to Artificial Intelligence*, R. Llagas, Ed. Amazon KDP Printing and Publishing, 2019, doi: 10.1007/978-3-319-98842-9_6.

[25] C. I. Nwakanma, M. S. Hossain, J.-M. Lee, and D.-S. Kim, "Towards machine learning based analysis of quality of user experience (QoUE)," *Int. J. Mach. Learn. Comput.*, vol. 10, no. 6, pp. 752–758, 2020.

LOVE ALLEN CHIJIOKE AHAKONYE (Member, IEEE) received the B.Sc. degree in mathematics/computer science from the University of Port Harcourt, Nigeria, in 2001, and the M.Sc. degree in information technology from the Federal University of Technology, Nigeria, in 2016. She is currently pursuing the Ph.D. degree with the Networked Systems Laboratory, Kumoh National Institute of Technology, Gumi, South Korea. She has over a decade of working experience in the nigerian oil and gas sector as a Network and System Administrator, from 2002 to 2016. From 2017 to 2019, she briefly worked as a Logistics Superintendent with Nigerian Petroleum Development Company. Since March 2021, she has been a full time Researcher with the Networked Systems Laboratory, Kumoh National Institute of Technology. Her research interests include AI-enabled energy clustering algorithms for smart factories and SCADA vulnerabilities and fault detection.

COSMAS IFEANYI NWAKANMA (Member, IEEE) received the Diploma degree (Hons.) in electrical/electronics engineering from Federal Polytechnic Nekede, Owerri, Imo State, Nigeria, in 1999, and the Bachelor of Engineering degree in communication engineering, the master's degree in information technology, and the Master of Business Administration (MBA) degree in project management technology from the Federal University of Technology, Owerri, in 2004, 2012, and 2016, respectively. He is currently pursuing the Ph.D. degree with the Networked System Laboratory, IT-Convergence Engineering, Kumoh National Institute of Technology, Gumi, South Korea. He has 12 years of lecturing and research experience at the Federal University of Technology. He was an intern with Asea Brown Boveri (ABB), Nigeria, in 2003. He is also a full time Researcher at the Networked System Laboratory, IT-Convergence Engineering, Kumoh National Institute of Technology. His research interests include reliability of artificial intelligence (AI), application to Internet of Things (IoT) for smart factories, homes, and vehicles.

JAE-MIN LEE (Member, IEEE) received the Ph.D. degree in electrical and computer engineering from Seoul National University, Seoul, South Korea, in 2005. From 2005 to 2014, he was a Senior Engineer with Samsung Electronics, Suwon, South Korea. From 2015 to 2016, he was a Principal Engineer at Samsung Electronics. Since 2017, he has been an Assistant Professor with the Department of IT-Convergence Engineering, School of Electronic Engineering, Kumoh National Institute of Technology, Gyeongbuk, South Korea. His current research interests include industrial wireless control networks, performance analysis of wireless networks, and TRIZ.

DONG-SEONG KIM (Senior Member, IEEE) received the Ph.D. degree in electrical and computer engineering from Seoul National University, Seoul, South Korea, in 2003. From 1994 to 2003, he worked as a full time Researcher at ERC-ACI, Seoul National University. From March 2003 to February 2005, he worked as a Postdoctoral Researcher at the Wireless Network Laboratory, School of Electrical and Computer Engineering, Cornell University, NY, USA. From 2007 to 2009, he was a Visiting Professor with the Department of Computer Science, University of California, Davis, CA, USA. He is currently the Director of the KIT Convergence Research Institute and ICT Convergence Research Center (ITRC and NRF Advanced Research Center Program) supported by Korean Government, Kumoh National Institute of Technology. His research interests include real-time IoT and smart platform, industrial wireless control networks, and networked embedded systems. He is a Senior Member of ACM.