Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.Doi Number

IMPACT: Impersonation attack detection via edge computing using deep autoencoder and feature abstraction

Seo Jin Lee[1], Paul D. Yoo[2], Senior Member, IEEE, A. Taufiq Asyhari[3], Senior Member, IEEE, Yoonchan Jhi[4], Lounis Chermak[1], Member, IEEE, Chan Yeub Yeun[5], Senior Member, IEEE, Kamal Taha[5], Senior Member, IEEE

[1]Centre for Electronic Warfare, Information and Cyber (CEWIC), Cranfield School of Defence and Security, Defence Academy of the United Kingdom, Shrivenham, SN6 8LA, United Kingdom
[2]Dept. CSIS, Birkbeck College, University of London, London, WC1E 7HX, United Kingdom
[3]School of Computing and Digital Technology, Birmingham City University, Birmingham B4 7XG, United Kingdom
[4]Security Research Center, Samsung SDS, Seoul, South Korea
[5]Center for Cyber-Physical Systems (C2PS), Khalifa University, Abu Dhabi, United Arab Emirates

Corresponding authors: Chan Yeob Yeun and Paul D. Yoo (e-mail: chan.yeun@ku.ac.ae and paul.d.yoo@ieee.org).

This work was supported in part by the Samsung's Global Outreach Fund under Grant P10458, and in part by the Center for Cyber-Physical Systems, Khalifa University, under Grant 8474000137-RC1-C2PS-T3.

ABSTRACT An ever-increasing number of computing devices interconnected through wireless networks encapsulated in the cyber-physical-social systems and a significant amount of sensitive network data transmitted among them have raised security and privacy concerns. Intrusion detection system (IDS) is known as an effective defence mechanism and most recently machine learning (ML) methods are used for its development. However, Internet of Things (IoT) devices often have limited computational resources such as limited energy source, computational power and memory, thus, traditional ML-based IDS that require extensive computational resources are not suitable for running on such devices. This study thus is to design and develop a lightweight ML-based IDS tailored for the resource-constrained devices. Specifically, the study proposes a lightweight ML-based IDS model namely IMPACT (IMPersonation Attack deteCTion using deep auto-encoder and feature abstraction). This is based on deep feature learning with gradient-based linear Support Vector Machine (SVM) to deploy and run on resource-constrained devices by reducing the number of features through feature extraction and selection using a stacked autoencoder (SAE), mutual information (MI) and C4.8 wrapper. The IMPACT is trained on Aegean Wi-Fi Intrusion Dataset (AWID) to detect impersonation attack. Numerical results show that the proposed IMPACT achieved 98.22% accuracy with 97.64% detection rate and 1.20% false alarm rate and outperformed existing state-of-the-art benchmark models. Another key contribution of this study is the investigation of the features in AWID dataset for its usability for further development of IDS.

INDEX TERMS IoT security, intrusion detection, feature engineering, mutual information, machine learning, edge computing.

I. INTRODUCTION

The role of edge devices has been elevated by the recent development of cloud and IoT technologies supporting in the need for intelligent, computing power and advanced services at the network edge. This new concept allows decentralised processes in interconnected devices. The rapid growth of interconnected smart and mobile devices has posed significant dangers on security and privacy of individuals, societies, nations and even in the extreme, the globe as a whole [1]. The impact of massive data breach and security threats is increasing with more advanced emerging applications such as healthcare, smart homes and cities and autonomous vehicles. All these domains deal with sensitive and confidential data, deeply mined from private activities on daily basis and the nature and scale of interconnection of the devices do not only seriously harm a single device or operator, however, all connected objects and involved humans in a large scale.

Such devices however have unique security challenges [2], among which their limited computational resources such as limited energy source (*e.g.* battery power) and computational power (*e.g.* processors and memories) [3][4]. The requirement of real-time processing also adds complexities on the development and deployment of both existing and new security measures.

IDS has been effective as a next line of defence for such computing devices and networks and extensively studied since the seminal work by Denning [5]. IDS can be classified into two major categories, signature and anomaly based. The anomaly-based IDS is designed to detect unknown attacks that deviate from the profile of normal network activities. On the other hand, signature-based system can only detect known attacks that can match the patterns or signatures stored in a database. As cyber-attacks are evolving, the flexibility and adaptability of signature-based IDS need to be further developed.

The concepts in machine learning (ML) and its subfield deep learning (DL) seem to be, by their inherent approaches, the right candidates for designing the adaptable IDS [6][7]. However, high-dimensional nature of ever-increasing data and iterative training process of models require extensive computational resources, thus, traditional ML-based IDS are not suitable for training and inference on resource-constrained devices.

Due to the high demand of computational resources for training and inference, the current approach is to transfer collected data to the central nodes (*e.g.* data centres) that have powerful resources. However, the distance between the devices and remote central nodes causes latency which could be a bottleneck to modern time-critical systems and applications that often require real-time processing of such big data. Besides, this centralised approach implies a single point of failure. For example, dysfunctionality or shutdown of a part of system leads to the failure of the entire system and has other issues including storage capacity, availability, scalability and privacy.

To mitigate the aforementioned problems, a new paradigm, called edge computing [8] has emerged. Its principle relies on the ability to perform computational tasks locally such as data processing and analysis are performed at the edge of the network near or at data sources rather than the central nodes. This paradigm benefits from the proximity between the data sources and computing nodes and also can solve the problem of poor or absent connectivity and bandwidth which are always required in the cloud-based systems. It is not surprising that cloud to the edge is one of the top strategic technologies for 2018 and 2020 according to a report by Gartner [9][10].

To compute efficiently and effectively closer to or at the edge of the network, the utilisation of ML approaches that can enable dimensionality reduction of data and efficient detection is critical. This study investigates potential ML methods to design and develop an efficient and effective ML-based IDS for the resource constrained edge devices which involve processing of a large amount of data and training of models. The key contributions of this study are:

- to determine the feasibility of a lightweight machine-learning IDS to be designed and deployed on resource constrained devices,
- to demonstrate, building upon earlier work [11][12], the effectiveness of extracted abstract features using a deep SAE, along with mutual information theoretic feature selection that outperforms other state-of-the-art models,
- to propose an architecture of gradient based SVM for the proposed IDS model,
- to analyse the temporal features within AWID dataset and their usability for the further development of IDS, and
- finally, to provide a new benchmark result on AWID dataset without using temporal features.

The remainder of this paper is organised as follows. Section II introduces the proposed IMPACT algorithm outlining its three novel concepts. Section III analyses and evaluates the performance of IMPACT and existing benchmark models along with investigations of the features of the AWID dataset and Section IV concludes with recommendations for further research.

II. RESEARCH METHODOLOGY

A. Data

To train, test and evaluate the proposed model, AWID dataset [13] was used due to its unique features in comparison with other existing datasets. While it contains new attack types, the AWID dataset is simulated using real-world wireless network which is a critical feature for modern IoT environments.

The dataset is divided by the types of attack classes. “ATK” set contains 16 attack classes and “CLS” has 4 classes in which 15 attacks are categorised by attack methodologies: impersonation, flooding and injection. For this study, “CLS” dataset is used and impersonation attack is considered only. The impersonation attacks included in the dataset are Caffe Latte, Evil Twin, and Hirte attack. Caffe Latte and Hirte are keystream retrieving attacks and Evil Twin is a man-in-the-middle attack according to their attack purpose. The tools used to implement the attacks include the Aircrack-ng suit, MDK3 tool, the Metasploit framework and custom tools made by authors using C language and the Lorcon2 library. Attackers mostly use the Airbase tool contained within Aircrack-ng suit for releasing Evil Twin attacks.

To gather the data, the authors created a realistic resource constrained environment of small office/home office (SOHO) wireless network infrastructure that consisted of a number of mobile and static clients such as smartphones, tablets, smart TV and laptops and a single mobile attacker node to release the attacks. A single Access Point (AP) was set up with the WEP encryption.

The dataset in the original form is imbalanced in such manner that the size of the normal class is significantly larger than the attack class with the ratio of 10:1 for the training set and 11:1 for the test set. Since this configuration could result in a bias during the model training phase, the dataset is balanced making the ratio 1:1 between the two classes for both training and test sets through pre-processing [12].

B. IMPACT

The IMPACT has three main components: i. feature extraction, ii. feature selection and iii. classification. Through feature extraction and selection, the dimensionality of data required for training and testing the model is reduced, increasing the efficiency of the model in terms of computational cost required to deploy on the resource-constrained devices. Stacked autoencoder (SAE), a type of deep neural network, was used for feature extraction and mutual information (MI) and C4.8 wrapper for feature selection. For the detection task, SVM with gradient descent optimisation was adopted that they were more effective in terms of detection performance compared to other models based on the experiment results.

To build the model, reduced AWID training and test datasets with 154 features were fed to the SAE. Through the SAE, a set of 50 new features with new data instances were extracted and appended to both the original training and test sets, producing the larger dataset with 204 features as a whole. This dataset was the input for the feature selection to find the reduced optimal feature subset and the reduced training and test sets with the final 5 selected features were used for training and testing of the ML classifier which produces the best classification result.

As shown in Fig. 1., an autoencoder (AE) [14] is a type of unsupervised neural network algorithm that learns from unlabeled data using backpropagation. It sets the output values to be the same as the input values, trying to learn the hypothesis function,

$$h_{W,b}(x) \approx x \tag{1}$$

The AE consists of an encoder and a decoder in which the encoder compresses input data into a low dimensional representation and the decoder reconstructs the input data from the low dimensional representation. In other words, the input data is replicated at the output layer. During the process of encoding, the input feature vectors are converted to an abstract feature vector and the dimensionality of the input data space can be reduced.

To achieve this, several constraints should be put on the network. For instance, setting the number of hidden neurons less than that of the input features, and some meaningful representations of the data can be discovered while attempting to reconstruct the input with the limited number of hidden neurons. Consequently, if some correlations exist between the features, the algorithm would be able to find them.

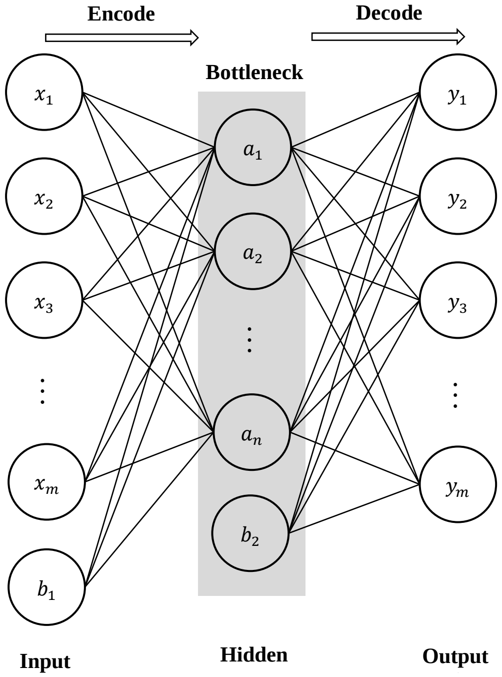


Fig. 1. Autoencoder (AE), where *m* and *n* indicate the number of neurons in the layer, *x* is an input feature, *b* is a bias, *a* is an activation and *y* is an output.

The constraint (2) is imposed on the hidden neurons in the encoders to compress the representation of the input data and extract features, where $\hat{\rho}_j$ (3) is the average activation and $a_j(x)$ is the activation of the hidden neuron j respectively. If the activation of the neuron j is 1, the neuron is active and if the activation is 0 (or -1 if $tanh$ is used as activation function instead of $sigmoid$), the neuron is inactive. The variable ρ denotes the sparsity parameter and is set to the value near $zero$ to force the neurons to be inactive most of the time.

$$\hat{\rho}_j = \rho \tag{2}$$

$$\hat{\rho}_j = \frac{1}{m}\sum_{i=1}^{m}\left[a_j(x)\right] \tag{3}$$

The cost function of the AE is specified by the mean squared error (MSE) function (4), given m training instances and the cost function for a single instance $\frac{1}{2}\left\|h_{W,b}\left(x^{(i)}\right) - y^{(i)}\right\|^2$.

$$MSE = \frac{1}{m}\sum_{i=1}^{m}\frac{1}{2}\left\|h_{W,b}\left(x^{(i)}\right) - y^{(i)}\right\|^2 \tag{4}$$

L2 regularisation (5), also called weight decay term, is added to the cost function, which will prevent overfitting by reducing the magnitude of the weights $W_{ji}^{(l)}$ between neuron *i* in layer *l* and neuron *j* in layer *l+1*:

where L is the total number of layers in the network and n and k are the number of neurons in layer l and $l+1$ respectively.

In addition, a penalty term, called sparsity regularisation (6) is added to the cost function to penalize $\hat{\rho}_j$ that diverges from ρ using the Kullback-Leibler (KL) divergence [15]. KL is a measure of the different between two different distributions. This function has the value either zero if (2) is satisfied or higher if $\hat{\rho}_j$ diverges from ρ. Hence, minimising this term encourages $\hat{\rho}_j$ to be close to ρ. S_2 is the number of hidden neurons within the encoder.

$$\Omega_{Sparsity} = \sum_{j=1}^{s_2} KL(\rho \parallel \hat{\rho}_j) = \sum_{j=1}^{s_2} \rho \log \frac{\rho}{\hat{\rho}_j} + (1-\rho) \log \frac{1-\rho}{1-\hat{\rho}_j} \quad (6)$$

The overall cost function is then the sum of MSE, L2 regularisation and sparsity regularisation term, where λ and β controls the strength of L2 regularisation and sparsity respectively.

$$J_{Sparse}(W,b) = MSE + \lambda * \Omega_{L2Reg} + \beta * \Omega_{Sparsity} \quad (7)$$

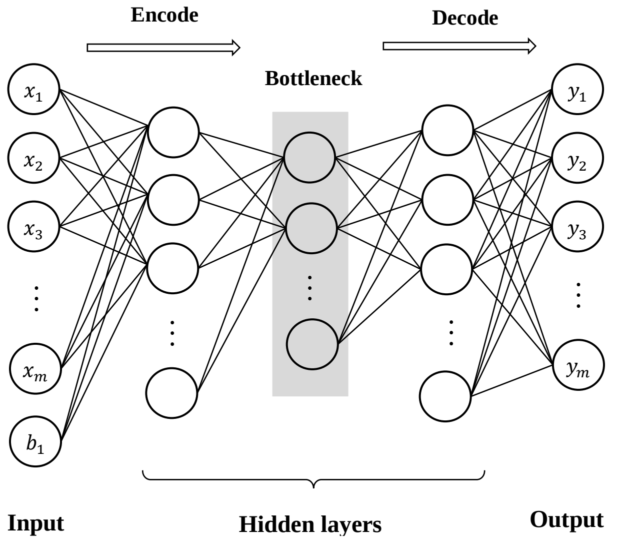


Fig. 2. Stacked autoencoder (SAE)

A stacked (or deep) autoencoder (SAE) consists of multiple AEs connected from one layer to the subsequent layer. The output of the previous encoder is the input of the next encoder and from this structure, higher representations, *i.e.* features, of the input data can be found. The reason why the SAEs was chosen is explained by the fact that a single AE behaves too greedily and important information for accurate classification of the target class could be discarded.

$$\Omega_{L2Reg} = \frac{1}{2} \sum_{l=1}^{L-1} \sum_{i=1}^{n} \sum_{j=1}^{k} (W_{ji}^{(l)})^2 \quad (5)$$

The SAE prevents such behaviour by refining gradually the neurons in the hidden layers. In other words, the SAE learns a better representation of the input data than a single AE. However, as more encoders need to be trained, the training time and complexity of model are increased. For the number of hidden neurons for two encoder layers, 100 and 50 were chosen respectively which were found to be optimal for the AWID impersonation dataset according to Aminanto *et al.* [11].

Following feature extraction using SAE, IMPACT performs feature selection to find the optimal feature subset from the whole feature set comprising of the original and extracted features produced from the feature extraction stage. This process finds the most relevant features and removes irrelevant features so that it reduces the complexity and computational cost of the model and also improves the detection performance. Hence, the feature selection can make the model both efficient and effective achieving the aim of this study. Among a variety of available methods, this study utilises mutual information (MI) and C4.8 wrapper.

MI is a quantity that measures the mutual dependence between two random variables. That is how much information one random variable has about another. In other words, it is the indication of the reduction in uncertainty of one random variable when given the knowledge about another. MI is related to the concept of entropy H (8) which is the expected information content in a random variable X:

$$H(X) = -\sum_{i} P(x_i) log P(x_i). \quad (8)$$

Herein P denotes the probability that an event with index i occurs. Conditional entropy (9) of two random variables X and Y with values x_i and y_j can be defined as

$$H(X|Y) = -\sum_{i,j} P(x_i, y_j) log \frac{P(x_i, y_j)}{P(y_j)} \quad (9)$$

where $P(x_i, y_j)$ is the joint probability distribution. Then, the definition of MI of two discrete variables X and Y is given by

$$\begin{aligned} I(X;Y) &= H(X) - H(X|Y) \\ &= H(X) + H(Y) - H(X,Y) \\ &= \sum_{i,j} P(x_i, y_j) log \frac{P(x_i, y_j)}{P(x_i)P(y_j)} \end{aligned} \quad (10)$$

where $H(X,Y)$ is the joint entropy. The higher the MI value is, less the uncertainty in a variable is and vice versa. Zero MI means the variables are independent.

C4.8 wrapper [16] is a decision tree-based algorithm extended from ID3. It uses pruning strategies to avoid overfitting. During the learning process of C4.8 algorithm, a decision tree is built first from the given training set using ID3, and then the learnt tree is converted into a set of rules, each of which is a rule for the path from the root to a leaf node. Each rule is pruned where preconditions that improve the estimated accuracy are removed. The pruned rules are then sorted by the accuracy and considered when subsequent instances are classified. A feature is useful for generalisation if it is present as a node or part of the rules and in contrast, the removed features are not important if they do not improve the accuracy. C4.8 utilises the measure of information gain (IG), which is exactly the MI, to select features and these features are then used as a subset for ML classifiers. Finally, IMPACT classifies network data into two classes using support vector classifier: normal and attack. For this task, linear support vector machine (SVM) with gradient descent as the optimiser is utilised.

$$c\big(x,y,f(\mathbf{x}_i)\big) = 1 - y_i f(\mathbf{x}_i) \tag{16}$$

$$c\big(x,y,f(\mathbf{x}_i)\big) = \begin{cases} 0, & if\ y_i f(\mathbf{x}_i) \geq 1 \\ 1 - y_i f(\mathbf{x}_i), & otherwise \end{cases} \tag{17}$$

Linear SVM is a supervised machine learning algorithm used to deal with binary classification problems that have two classes. Many possible boundaries or hyperplanes that can separate the classes exist, thereby a method to find the best one is required. SVM aims to find the optimal decision boundary (or maximum-margin hyperplane) in the way that the margin between the boundary and the nearest data instances of the classes is maximised as shown in Fig. 3. The nearest data instances that define the maximum margin (or hyperplane) are called support vectors.

Given a training data of n instances $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)$, where y_i is the true class of input data $\mathbf{x}_i$ $(i = 1, \ldots, n)$ and either 1 or -1, the decision boundary is defined as

$$f(\mathbf{x}_i) = \mathbf{w}^{\mathrm{T}}\mathbf{x}_i + b = 0 \tag{11}$$

where $\mathbf{w}$ is the weight vector and b is the bias.

To prevent the data instances from lying on the incorrect side, the following constraints are added for each i:

$$if\ y_i = 1, \qquad \mathbf{w}^{\mathrm{T}}\mathbf{x}_i + b \geq 1 \tag{12}$$

$$if\ y_i = -1, \qquad \mathbf{w}^{\mathrm{T}}\mathbf{x}_i + b \leq -1 \tag{13}$$

and these can be combined into

$$y_i(\mathbf{w}^{\mathrm{T}}\mathbf{x}_i + b) \geq 1\ for\ all\ 1 \leq i \leq n. \tag{14}$$

SVM can solve non-linearly separable problems by utilising the method called kernel trick that maps the original data into higher dimensional space to make the data linearly separable. A potential limitation is that SVM may require extensive training time. Though SVM produced high performance results, the training times are often too high in comparison to other classifiers. However, in this study, by using a linear form of SVM, the training time was reduced while achieving comparable results.

SVM uses hinge loss as its loss function for optimisation. In linear SVM, for an output $y_i = \pm 1$, the hinge loss can be defined as

$$max(0, 1 - y_i f(\mathbf{x}_i)). \tag{15}$$

If $f(\mathbf{x}_i)$ predicts the correct class, then y_i and $f(\mathbf{x}_i)$ have the same sign and $y_i f(\mathbf{x}_i) \geq 1$, so the loss is zero. If y_i and $f(\mathbf{x}_i)$ have the opposite sign and $y_i f(\mathbf{x}_i) < 1$, the loss increases linearly. The hinge loss penalizes incorrect classifications within $y_i f(\mathbf{x}_i) < 1$ that corresponds to a margin in SVM.

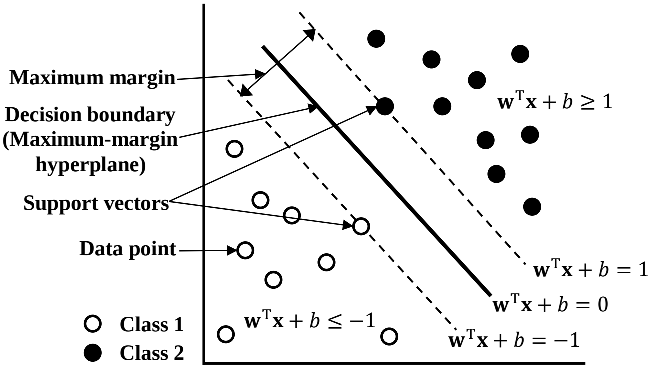

Maximum margin
Decision boundary (Maximum-margin hyperplane)
Support vectors
Data point
○ Class 1
● Class 2
$\mathbf{w}^{\mathrm{T}}\mathbf{x} + b \geq 1$
$\mathbf{w}^{\mathrm{T}}\mathbf{x} + b = 1$
$\mathbf{w}^{\mathrm{T}}\mathbf{x} + b = 0$
$\mathbf{w}^{\mathrm{T}}\mathbf{x} + b \leq -1$
$\mathbf{w}^{\mathrm{T}}\mathbf{x} + b = -1$

Fig. 3. Linear Support Vector Machine

The objective function $J(\mathbf{w})$ (18) consists of two terms: regularisation term and loss. As the hinge loss function is convex, ML convex optimisers can be used. For optimisation, the objective function should be minimised:

$$Minimize\ J(\mathbf{w}) = \frac{\lambda}{2}\|\mathbf{w}\|^2 + \frac{1}{n}\sum_{i=1}^{n} max(0, 1 - y_i f(\mathbf{x}_i)) \tag{18}$$

Gradient descent takes steps iteratively to update parameters in the direction of the gradient. To run gradient descent, derivatives with respect to b and $\mathbf{w}$ are required. However, the hinge loss is not differentiable, thus, a sub-gradient should to be used with respect to $\mathbf{w}$ and $f(\mathbf{x}_i)$ as follows:

$$\frac{\partial}{\partial \mathbf{w}} max(0, 1 - y_i f(\mathbf{x}_i)) = \begin{cases} 0, & if\ y_i f(\mathbf{x}_i) \geq 1 \\ -y_i \mathbf{x}_i, & otherwise \end{cases} \tag{19}$$

III. EVALUATION AND ANALYSIS

The confusion matrix is commonly used to evaluate the performance of a ML model, particularly for binary classification which is the case in this study. Based on the confusion matrix, the below evaluations measures are intended to give information on the effectiveness and efficiency of the proposed algorithm. The evaluation measure used are accuracy (Acc), detection rate (DR), false alarm rate (FAR), F-measure (F$_1$), Mathew's correlation coefficient (Mcc) and Time To Build (TTB) and can be calculated using the below equations.

$$Acc = \frac{TP + TN}{TP + TN + FP + FN} \tag{20}$$

$$DR(recall) = \frac{TP}{TP + FN} \tag{21}$$

$$Precision = \frac{TP}{TP + FP} \tag{22}$$

$$FAR = \frac{FP}{TN + FP} \tag{23}$$

$$FNR = \frac{FN}{FN + TP} \tag{24}$$

$$F_1 = \frac{2TP}{2TP + FP + FN} \tag{25}$$

$$Mcc = \frac{(TP * TN) - (FP * FN)}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{26}$$

A. Theoretic Feature Selection using Mutual Information

After the feature extraction process, MI values for all 204 features consisting of original 154 and extracted 50 features were calculated. The features then were ranked from the highest to the lowest MI values. Among 204 features, 83 features were found to have the MI values greater than 0 and the rest 121 features had the value 0 which means that they had no relevance to the attack class. All 50 extracted features were among the afore-mentioned 83 features whereas 33 were original features of which only 4 features were within top 20 features. This suggests that the SAE was able to successfully extract the features that are relevant to the attack class with meaningful representations. In turn, it demonstrates the effectiveness of SAE as a feature extraction method to build a lightweight IDS by discovering relatively more meaningful features and reducing the dimensionality of data and the complexity of the model.

Among the most relevant 20 features based on the MI values were original features 4, 7, 8, 9, 38 and 82, however, there was some redundancy that the features 4 and 7 had exactly the same data instances resulting in the same MI values and so did 8 and 9. Therefore, features 7 and 9 were removed from the datasets for training the model. The top 20 features based on MI values are 8, 82, 4, 38, 157, 162, 168, 160, 188, 161, 199, 176, 159, 191, 182, 186, 195, 156, 158 and 165 [12].

To find the optimal subset from the top 20 features, Parker *et al.* [6] experimented five wrapper algorithms to select features and evaluated in terms of the number of features and Acctraining time with the aim of minimising computational cost for resource-constrained devices. C4.8 has taken the least time compared to the other algorithms. In terms of the number of features, C4.8 had only one or two more features than RF, MLP, and RBF that were significantly slower than C4.8 even though they resulted in the smaller number of features. Though logistic regression was the second fastest algorithm, it had the number of features twice or more than all the other algorithms, significantly increasing the complexity and computational cost of the model. The selected feature subset consists of five features including three original features 4, 8 and 82 and two extracted features 156 and 157.

B. Gradient-based Optimisation

The weights of SVM are found using gradient decent algorithm. Learning rate of 0.00001 achieved the highest DR and lowest FNR, however, it showed the worst performance in Acc, Precision, FAR, F_1 and Mcc. There is a trade-off between DR and FAR as DR tends to fall whereas FAR improves. The overall performance slightly improves between 0.00001 and 0.1 and rapidly increases between 0.1 and 0.5. Acc, FAR, Precision, F_1 and Mcc gradually increase until 0.1 then rapidly improves until 0.5. Therefore, the learning rates around 0.5 – 0.51 and 0.52 – were investigated. In addition to that, 0.5 has the highest Acc, F_1 and Mcc, however, also the second highest Precision and second lowest FAR and DR, thus, 0.5 was chosen. Learning rate of 0.55 has the highest Precision and lowest FAR, however, worse in other metrics than 0.5. The final results using learning rate of 0.5 are provided in Table I.

C. Comparisons between baselines and IMPACT

The most recent research for impersonation attack using AWID datasets were performed by Kolias *et al.* [13], Aminanto *et al.* [17], D-FES Corr [11] and DEMISe-RBFC and DETEReD [12]. As shown in Table I, IMPACT achieved the highest F_1 and Mcc while Acc is the second highest and FAR is the second lowest. Kolias *et al.* [13] has the lowest Mcc and the highest FAR. This is considered to be due to the imbalanced dataset used and feature selection method that Kolias *et al.* [13] utilised only expert knowledge without any ML, data-driven or statistical methods. Compared to D-FES Corr [11], though IMPACT has higher FAR by 0.16% and lower Acc by 0.004%, it achieved higher DR by 1.73%, higher F_1 by 2.04% and higher Mcc by 1.4%.

TABLE I
COMPARISONS BETWEEN IMPACT AND THE STATE-OF-THE-ART MODELS

Model	No of features	Acc	DR	FAR	F_1	Mcc	TTB
IMPACT	**5 (3+2)**	98.22	97.64	01.20	**98.21**	**96.45**	299.97*
DEMISe-RBFC [12]	7 (4+3)	98.00	99.04	03.00	97.98	96.02	301.53*
DETEReD [12]	**5 (3+2)**	98.04	**99.07**	02.96	98.01	96.09	603.33**
D-FES-Corr [11]	12	**98.26**	95.91	**01.04**	96.17	95.05	1264.00
Kolias *et al.* [13]	20	94.91	97.23	74.21	97.37	22.12	NRA
Aminanto *et al.* [17]	35	97.60	85.00	02.36	NRA	NRA	NRA

The performance of IMPACT is measured on learning rate of 0.5 on the feature subset of three original features (4, 8 and 82) and two abstract features (156 and 157) only.
* The time to build (TTB) for the models includes the 293s required for SAE.
** Includes TTB required for both SAE and C4.8 wrapper.
NRA = No results available.

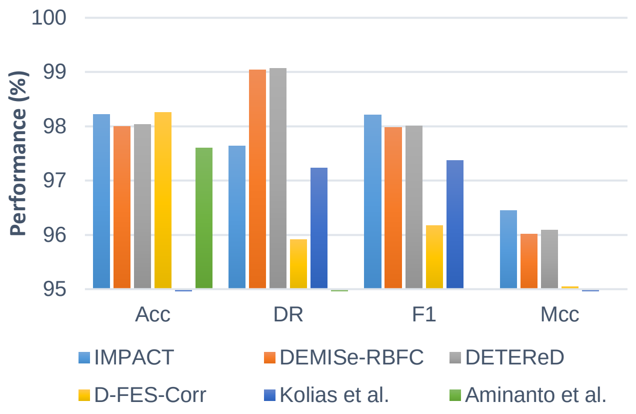


Fig. 4. Performance comparisons between IMPACT and the state-of-the-art-models in terms of Acc, Dr, F1 and Mcc (Kolias *et al.*'s Acc and Mcc and Aminanto *et al.*'s DR, F1 and Mcc are out of range.)

Even though DETEReD and DEMISe-RBFC achieved the highest DR and excelled in Acc, F_1 and Mcc, DETEReD has the highest FAR and DEMISe-RBFC the second highest. Both have much higher FAR—more than double—than either IMPACT or D-FES Corr [11]. Considering the throughput of network data in the era of big data, this amount of false alarms cannot be ignored because it will cause much higher cost to network administrators than IMPACT. Within the context of IDS, minimising FAR is crucial. In comparison with DETEReD and DEMISe-RBFC, the IMPACT has lower DR, however, it is still higher than those of three other models (D-FES Corr, Kolias *et al.* and Aminanto *et al.*) and has FAR less than half of the results of DETEReD and DEMISe-RBFC. The reason why DETEReD had the better result for DR is that it had more number of TP than that of IMPACT whereas IMPACT had a higher sum of TP and TN for Acc than that of DETEReD. The values of the denominators for both DR and Acc were the same in the two models. For Mcc, IMPACT had a higher proportion of the numerator per denominator than DETEReD.

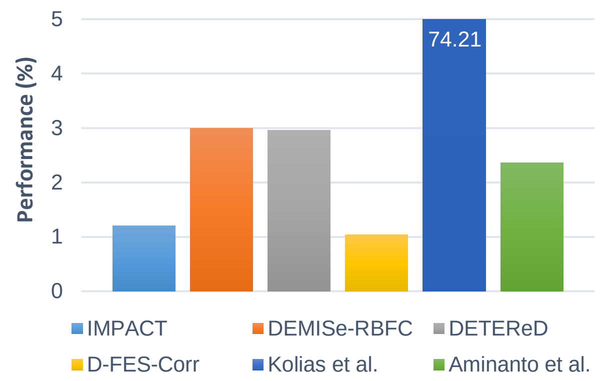


Fig. 5. Performance comparisons between IMPACT and the state-of-the-art-models in terms of FAR (Kolias *et al.* is out of range.)

IMPACT performed better with the optimised subset selected, using C4.8, from the top 20 features rather than 10 features in contrast to the result produced by DEMISe in which the authors' logistic regression classifier showed better performance with the optimised subset from the 10 features.

The training time of the model is also an important measure for computational time efficiency of the model. IMPACT has TTB requirement considering of SAE and classifier training time, but excluding the time required for C4.8 wrapper as the feature subset was provided by the authors of the earlier work, DEMISe, and there was no need of re-running C4.8. Kolias *et al.* [13] and Aminanto *et al.* [17] do not provide the exact model build time. All the models were run on different hardware setups, thus, the models cannot have fair comparison in terms of training time. However, in terms of the number of features which could be a measure of memory efficiency of the model, IMPACT utilises the least number of features, significantly less than the three other benchmark models (D-FES, Kolias *et al.* and Aminanto *et al.*), while outperforming them.

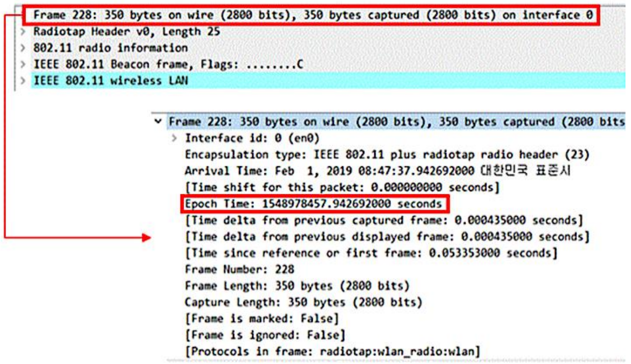
Overall, the IMPACT achieved the performance t mitigates the drawbacks of DETEReD and D-FES Corr as FAR is significantly lower than DETEReD, DR is better than D-FES Corr and F_1 and Mcc are the best among all the other models. Based on the evaluation of the comparison of performance results, it proves the effectiveness of SAE, MI and C4.8 wrapper methods for the dimensionality reduction of dataset for the lightweight IDS reducing computational cost in terms of time and space.

TABLE II
SELECTED FEATURES USING C4.8

Feature	Name	Description
4	frame.time_epoch	Epoch time when this frame was captured
8	frame.len	Frame length on the wire
82	wlan.seq	Sequence number
156	Extracted feature	Extracted from SAE
157	Extracted feature	Extracted from SAE

D. AWID Feature Analysis

Each feature within AWID dataset has been investigated in order to verify if any of them contains temporal information. The temporal features (*a.k.a.* time domain features), which are simple to extract and have an easy physical interpretation. However, if the presence of information within the temporal


```
Frame 228: 350 bytes on wire (2800 bits), 350 bytes captured (2800 bits) on interface 0
› Radiotap Header v0, Length 25
› 802.11 radio information
› IEEE 802.11 Beacon frame, Flags: ........C
› IEEE 802.11 wireless LAN

˅ Frame 228: 350 bytes on wire (2800 bits), 350 bytes captured (2800 bits
  › Interface id: 0 (en0)
    Encapsulation type: IEEE 802.11 plus radiotap radio header (23)
    Arrival Time: Feb  1, 2019 08:47:37.942692000 대한민국 표준시
    [Time shift for this packet: 0.000000000 seconds]
    Epoch Time: 1548978457.942692000 seconds
    [Time delta from previous captured frame: 0.000435000 seconds]
    [Time delta from previous displayed frame: 0.000435000 seconds]
    [Time since reference or first frame: 0.053353000 seconds]
    Frame Number: 228
    Frame Length: 350 bytes (2800 bits)
    Capture Length: 350 bytes (2800 bits)
    [Frame is marked: False]
    [Frame is ignored: False]
    [Protocols in frame: radiotap:wlan_radio:wlan]
```

Fig. 4. Feature 4 Epoch Time in Wireshark

features are used for the learning algorithm, then it will remember the time of each attack during the training. Consequently, detected observations in testing set rely on temporal information learned from training set. That being said, this information must not be used in training or in testing the model as in a real scenario an IDS never knows the actual time of attack.

In impersonation attack such as Evil Twin and Caffe Latte, it is found that the number of beacon frames in the victim's network are almost doubled and about half of these frames contained intrusive characteristics, that is, the impersonation attacks occurred during these durations [13].

Unfortunately, we found that Kolias *et al.* [13] set up an attacker and attacks were injected at particular times in their experiments and these were recorded in some of their features. The features used in AWID dataset were derived from Wireshark and the full list can be found in the official AWID dataset website [18] and Wireshark display filter reference page [19]. Among the selected features in Table II, the top ranked raw feature 4 (*frame.time_epoch*) is the epoch time when the frame was captured as shown in Fig. 4 and the redundant feature 7 (*frame.time_relative*) also has the same characteristics as feature 4 and therefore, it had the same MI value as feature 4 as mentioned in Section III-A. Additionally, feature 38 (*radiotap.mactime*) is MAC timestamp, another temporary feature, defined in Radiotap [20] as "Value in microseconds of the MAC's 64-bit 802.11 Time Synchronization Function timer when the first bit of the MPDU arrived at the MAC. For received frames only." [20]. We found that the benchmark models, DEMISe and DETEReD utilised the temporal feature of 4, while Aminanto *et al.* and D-FES utilised all three temporal features of 4, 7, and 38.

TABLE III
PERFORMANCE RESULTS OF IMPACT WITHOUT TEMPORAL FEATURES

Acc	DR	FAR	F_1	Mcc
94.72	94.04	04.61	94.68	89.43

As the final selected feature set of IMPACT has only temporal feature (*frame.time_epoch*), the model trained without the temporal feature was experimented and the results showed that the model without the feature had worse performance than the model with the feature. Therefore, it has been proved that the temporal feature significantly contributed to the performance of the model and the feature selection method was effective, however, in fact, this feature is not valid to use for the development of IDS.

IV. CONCLUDING REMARKS

This paper presents the development of a machine learning based approach of an IDS offering the ability to be deployed and run directly on the resource-constrained devices. This was achieved through a smart strategy aiming to reducing the complexity of the model which consists of two main steps. First to the ability reduce the number of features through feature extraction and selection using SAE and MI and to evaluate their effectiveness in both efficiency and performance. The results showed that the extracted abstract features were selected as top features among the whole set of original and extracted features. MI values of the features could be utilised to select most relevant features and remove irrelevant features, resulting in the reduction of the complexity of the model without decreasing the performance, however, outperforming other models.

The second step consisted in training and testing the linear SVM using gradient descent. In comparison with other models using different classifiers or SVM, (providing higher training time on the AWID impersonation dataset), the IMPACT demonstrated better performance including much lower FAR compared to DEMISe models. With the investigation of temporal features existing in AWID dataset, IMPACT provided its new benchmark results without using any temporal features in AWID dataset proving that it is the only ML-based IDS tailored for resource constrained devices and which is independent of such features in contrary to its competing DEMISe, DETEReD and D-FES algorithms.

Based on these findings, the ways for further development could be proposed. Firstly, successful use of an SAE, opens perspectives for the use of other deep neural networks to extract abstract features. Secondly, this study only focuses on impersonation attack, however, there are two other type of attacks in AWID dataset, flooding and injection. IMPACT has not been yet tested against these, neither on newer attack types found in wireless IoT networks. Finally, IMPACT needs to be trained and tested on additional datasets providing their own features existing within the IDS research in order to prove its usefulness and effectiveness. Today, most wireless sensor network used as an automatic data acquisition and transmission system in monitoring applications is based on 802.15.4. However, the dataset in [7] is built on 802.11. For the usefulness and coverage of the proposed algorithm, in our future work, the proposed algorithm will be tested on a new benchmark dataset created on 802.15.4.

ACKNOWLEDGMENT

This work was supported by the Samsung's Global Outreach Fund under the grant agreement No. P10458.

REFERENCES

[1] R. Ande, B. Adebisi, M. Hammoudeh, and J. Saleem, "Internet of Things: Evolution and technologies from a security perspective," *Sustain. Cities Soc.*, p. 101728, Jul. 2019.
[2] A. Mosenia and N. K. Jha, "A comprehensive study of security of internet-of-things," *IEEE Trans. Emerg. Top. Comput.*, vol. 5, no. 4, pp. 586–602, Oct. 2017.
[3] S. Raza, L. Wallgren, and T. Voigt, "SVELTE: Real-time intrusion detection in the Internet of Things," *Ad Hoc Networks*, vol. 11, no. 8, pp. 2661–2674, Nov. 2013.
[4] G. Choudhary, V. Sharma, I. You, K. Yim, I.-R. Chen, and J.-H. Cho, "Intrusion Detection Systems for Networked Unmanned Aerial Vehicles: A Survey," Jul. 2018.
[5] D. E. Denning, "An Intrusion-Detection Model," *IEEE Trans. Softw. Eng.*, vol. SE-13, no. 2, pp. 222–232, Feb. 1987.
[6] Y. Xin *et al.*, "Machine Learning and Deep Learning Methods for Cybersecurity," *IEEE Access*, vol. 6, pp. 35365–35381, 2018.
[7] A. Shenfield, D. Day, and A. Ayesh, "Intelligent intrusion detection

systems using artificial neural networks," *ICT Express*, May 2018.

[8] W. Yu *et al*., "A Survey on the Edge Computing for the Internet of Things," *IEEE Access*, vol. 6, pp. 6900–6919, 2018.

[9] K. Panetta, "Gartner Top 10 Strategic Technology Trends for 2018 - Smarter With Gartner," 2017. [Online]. Available: https://www.gartner.com/smarterwithgartner/gartner-top-10-strategic-technology-trends-for-2018/. [Accessed: 14-Nov-2018].

[10] K. Panetta, "Gartner Top 10 Strategic Technology Trends for 2020 - Smarter With Gartner," 2019. [Online]. Available: https://www.gartner.com/smarterwithgartner/gartner-top-10-strategic-technology-trends-for-2020/. [Accessed: 13-Nov-2019].

[11] M. E. Aminanto, R. Choi, H. C. Tanuwidjaja, P. D. Yoo, and K. Kim, "Deep Abstraction and Weighted Feature Selection for Wi-Fi Impersonation Detection," *IEEE Trans. Inf. Forensics Secur.*, vol. 13, no. 3, pp. 621–636, Mar. 2018.

[12] L. R. Parker, P. D. Yoo, T. A. Asyhari, L. Chermak, Y. Jhi, and K. Taha, "DEMISe: Interpretable Deep Extraction and Mutual Information Selection Techniques for IoT Intrusion Detection," in *Proceedings of the 14th International Conference on Availability, Reliability and Security*, 2019, pp. 98:1--98:10.

[13] C. Kolias, G. Kambourakis, A. Stavrou, and S. Gritzalis, "Intrusion Detection in 802.11 Networks: Empirical Evaluation of Threats and a Public Dataset," *IEEE Commun. Surv. Tutorials*, vol. 18, no. 1, pp. 184–208, 2016.

[14] "Unsupervised Feature Learning and Deep Learning Tutorial." [Online]. Available: http://ufldl.stanford.edu/tutorial/unsupervised/Autoencoders/. [Accessed: 01-Mar-2020].

[15] S. Kullback and R. A. Leibler, "On Information and Sufficiency," *Ann. Math. Stat.*, vol. 22, no. 1, pp. 79–86, Mar. 1951.

[16] J. R. Quinlan, *C4.5: Programs for Machine Learning*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1993.

[17] M. E. Aminanto and K. Kim, "Detecting impersonation attack in wifi networks using deep learning approach," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2017, vol. 10144 LNCS, pp. 136–147.

[18] "List of Attributes - Wireless Security Datasets Project." [Online]. Available: http://icsdweb.aegean.gr/awid/attributes.html. [Accessed: 23-Sep-2019].

[19] "Wireshark · Display Filter Reference: Frame." [Online]. Available: https://www.wireshark.org/docs/dfref/f/frame.html. [Accessed: 23-Sep-2019].

[20] "Radiotap - TSFT." [Online]. Available: https://www.radiotap.org/fields/TSFT.html. [Accessed: 26-Feb-2020].

SEO JIN LEE received the BSc degree in Computing from University College London (UCL), United Kingdom in 2017. She is currently pursuing the MSc by Research degree at Centre for Electronic Warfare, Information and Cyber (CEWIC) within Cranfield School of Defence and Security based in the Defence Academy of the United Kingdom, Shrivenham. Her research interests focus on data-driven intelligence including big data analytics and machine learning techniques applied to cyber defence and security problems.

PAUL D. YOO is currently with the CSIS within Birkbeck College at the University of London leading BIDA Data-driven Cyber Security Laboratory. Prior to this, he held academic/research posts in Cranfield (Defence Academy of the UK), Sydney (USyd) and South Korea (KAIST). In his career, he has amassed more than 80 prestigious journal and conference publications, has been awarded more than US$ 2.3 million in project funding, and a number of prestigious international and national awards for his work in advanced data analytics, machine learning and secure systems research, notably IEEE Outstanding Leadership Award, Capital Markets CRC Award, Emirates Foundation Research Award, and the ICT Fund Award. Most recently, he won the prestigious Samsung award for research to protect IoT devices using machine-learning approach and Research England's Global Challenge Research Fund (GCRF) for research to protect global environment (e.g. marine resources) using edge intelligence techniques. Paul currently serves as an Associate Editor for IEEE Transactions on Sustainable Computing, IEEE Access and Journal of Big Data Research (Elsevier). He had served as an Editor for IEEE COMML (big data and machine learning areas) from 2014 to 2019. He is also affiliated with the University of Sydney and Korea Advanced Institute of Science and Technology (KAIST) as a Visiting Professor. Paul is a Senior Member of the IEEE and a Fellow of HEA.

A. TAUFIQ ASYHARI received the B.Eng. degree (First Class Honors) in Electrical and Electronic Engineering from Nanyang Technological University (NTU), Singapore, in 2007, and the Ph.D. degree in Information Engineering from the University of Cambridge, U.K., in 2012. He is currently an Associate Professor (eqv. to Reader) in Networks and Communications with the School of Computing and Digital Technology with Birmingham City University, UK. He holds visiting appointments at Cranfield University, U.K. and Telkom University, Indonesia. He previously held full-time/visiting positions at National Chiao Tung University (NCTU), Bell Laboratories and the University of Stuttgart. His research interests include the areas of information theory, communication and coding theory, and signal processing techniques with applications to wireless and molecular communication networks, Internet of Things, and data analytics. Dr. Asyhari is a Fellow of the Higher Education Academy, U.K., and a Senior Member of the IEEE. He has received the notable Samsung Global Research Outreach Award, in 2017, the Silver Medal at the International Trade Fair iENA 2017, the IEEE EURASIP Best Paper Award from the 11th IEEE—ISWCS 2014, Barcelona, Spain, the National Science Council of Taiwan Starting Grant, in 2013, and the Cambridge Trust – Yousef Jameel Scholarship. He has served as the Demo/Poster/WiP Chair of the IEEE DASC 2018, Greece, and was the Session Chair of the IEEE SSCI 2016, Greece, and the IEEE ISIT 2013, Turkey. He has served as a member of the Technical Programme Committee in a number of leading international conferences. He has been frequently invited to review articles in prestigious journals and conferences. Since 2018, he has been an Associate Editor of IEEE ACCESS journal.

YOONCHAN JHI received the BE and ME Degrees in Computer Science and Engineering from Inha University, South Korea, and the Ph.D. degree in Computer Science and Engineering from the Pennsylvania State University, University Park, in 2010. He is currently leading the Security Research Center at Samsung SDS, Seoul, South Korea. His research interests include programming languages, software analysis, network security, and systems security. His current work focuses on IoT and cyber-physical systems-related security.

LOUNIS CHERMAK (M'12) received his Engineering Degree in Electrical Engineering from Polytech' Grenoble, Grenoble, France, in 2010. He then obtained an MSc. and Ph.D. degrees in Computer Vision from Cranfield University, Cranfield - Shrivenham, UK, in 2011 and 2014 respectively. From 2014 to 2017 he held a position of Research Fellow. Since 2017, he is Assistant Professor in Computer Vision and Autonomous Systems at Cranfield University, Shrivenham, UK, where he currently leads the Autonomy Group and the Autonomous Systems Laboratory, in the Centre of Electronic Warfare Information and Cyber, at School of Defence and Security. He has authored and co-authored several technical papers and has been reviewer for high impact journals and conferences. His research interests include development and implementation of techniques and algorithms for visual based aerial, ground and space applications towards perception and autonomous and mobility. He is a recipient of the Selwyn Award, from the Royal Photography Society in 2017. He also a technical member of the NATO Science and Technology Organization where he has been Co-Chairing an Exploratory Team and a Research Task Group activity on Autonomous Vehicle Technology since 2018.

CHAN YEOB YEUN holds M.Sc. (1996) and Ph.D. (2000) degrees in Information Security from Royal Holloway, University of London. After his Ph.D., he joined Toshiba TRL in Bristol, UK, and later became a Vice President at LG Electronics, Mobile Handset R&D Center, in Seoul, Korea (2005). He was responsible for developing mobile TV technologies and related security. He left LG Electronics in 2007 and joined KAIST (Korea) until August 2008 and then Khalifa University Science and Technology in September 2008. He is currently a researcher in cybersecurity including IoT/USN security, cyber-physical system security, cloud/fog security and cryptographic techniques as an Associate Professor in the Department of Electrical Engineering and Computer Science and an active member of Center for Cyber-Physical Systems (C2PS). He also enjoys lecturing for M.Sc. Information Security and Ph.D. Engineering courses at Khalifa University. He has published more than 130 journal articles and conference papers, 9 book chapters and 10 international patent applications. He also serves on the editorial board of multiple international journals and on the steering committee of international conferences. He is a senior member of the IEEE.

KAMAL TAHA is an Associate Professor in the Department of Electrical and Computer Engineering at Khalifa University, UAE, since 2010. He received his Ph.D. in Computer Science from the University of Texas at Arlington, USA. He has over 90 refereed publications that have appeared in prestigious top ranked journals, conference proceedings, and book chapters. Over 20 of his publications have appeared in IEEE Transactions journals. He was as an Instructor of Computer Science at the University of Texas at Arlington, USA, from August 2008 to August 2010. He worked as Engineering Specialist for Seagate Technology, USA, from 1996 to 2005 (Seagate is a leading computer disc drive manufacturer in the US). His research interests span bioinformatics, Information Forensics & Security, information retrieval, data mining, databases, and *defect* characterization of *semiconductor wafers*, with an emphasis on making data retrieval and exploration in emerging applications more effective, efficient, and robust. He serves as a member of the Program Committee, editorial board, and review panel for a number of international conferences and journals, some of which are IEEE and ACM journals. He is a Senior Member of the IEEE.