Enhancing File Entropy Analysis to Improve Machine Learning Detection Rate of Ransomware

CHIA-MING HSU, CHIA-CHENG YANG, HAN-HSUAN CHENG, PAUL E. SETIASABDA, AND JENQ-SHIOU LEU, (Senior Member, IEEE)
Department of Electronic and Computer Engineering, National Taiwan University of Science and Technology, Taipei 10607, Taiwan
Corresponding author: Jenq-Shiou Leu (jsleu@mail.ntust.edu.tw)
This work was supported by the Ministry of Science and Technology, Taiwan, under Grant MOST-109-2221-E-011-108.

ABSTRACT Cybersecurity is the biggest threat in the world. More and more people are used to storing personal data on a computer and transmitting it through the Internet. Cybersecurity will be an important issue that everyone continues to pay attention to. One of the most serious problems recently is the prevalence of ransomware, especially crypto-ransomware. Unlike ordinary attacks, crypto-ransomware does not control the victim’s computer and steal important data. It focuses on encrypting all data and asking victims to provide ransom to decrypt the data. Currently, many studies focus on various aspects of ransomware, including file-based, behavior-based, and network-based ransomware detection method, and use machine learning to build detection models. In addition to the above research, we found that attackers have begun to develop a new method to encrypt data. It will not only increase the speed of data encryption but also reduce the detection rate in the existing detection system. In any case, we are still facing ransomware dangers, as it is hard to recognize and forestall ransomware executing obscure malicious programs. In other words, user data will be sabotaged as soon as the computer cannot detect the ransomware. To solve the problem, detecting files instead of detecting the executable program might be helpful to establish the backup system immediately before ransomware encrypts all of the user files. We analyze the 22 formats of the encrypted files, extract the specific features and use the Support Vector Machine to distinguish between encrypted and unencrypted files. Conducted analysis results confirm that our method has better performance and a higher detection rate, reaching 85.17%. (Where the detection rate of SVM kernel Trick (Poly) exceeds 92%).

INDEX TERMS Machine learning, ransomware, entropy, security.

I. INTRODUCTION According to Trend Micro Security Company research, ransomware which was evaluated from fake AV to crypto-ransomware have continued to evolve in the past few years and have started to be used to attack specific targets, such as government agencies, hospitals, and big companies which are related to people’s daily life. Because those targets can’t afford the huge economic losses, they will pay the attacker to decrypt the files. In addition, the attacker will ask the victim to pay the ransom through cryptocurrency like bitcoin which is very difficult to trace back to find out the account owners. The ransomware RyuK attacked the US government agencies in 2019, making at least US $370 million for hackers from August 2018 to January 2019.

The associate editor coordinating the review of this manuscript and approving it for publication was Khin Wee Lai.

This low-risk and high-reward criminal industry is expected to continue to flourish.

There are many kinds of research that focus on analyzing, detecting, and preventing ransomware from damaging computers. But at the same time, the attackers also continue to develop new technologies to impede the analysis and detection of ransomware. The File-based detection method [1]–[3] focuses on detecting ransomware that has the same fixed patterns in file content or structure. The analysts need a long time to analyze and figure out the specific pattern which only the ransomware has, then transfer that into some detection rules like YARA for detecting the new malware. But the attacker used some methods like polymorphic, metamorphic, or obfuscated code to modify the malware easily and quickly to avoid being detected. Instead of detecting the fixed content of ransomware, the behavior-based detection method makes use of the specific activities which the
malware always takes in the computer like modifying register settings for auto-running after computer restart. Or search, open and encrypt the files by using the windows file I/O and encrypt API functions [4], [5]. But sometimes it causes a high false-positve rate because lots of benign software have the same behaviors mentioned above. The connection-based detection method monitors and detects the network traffic of the computer. For key exchange, to report the victims' information and receive the attacker's command, the ransomware needs to be connected to the command and control (C&C) server while executed. Each kind of ransomware has its own network protocol and the cybersecurity malware analyst can analyze and extract the specific patterns from the malicious traffic and detect it. But the attacker began to use the standard network protocol to transmit the malicious traffic, like using Dropbox as a C&C server, the network traffic will be encrypted by standard SSL protocol. Some researchers found that the ransomware has a special order while encrypting files, so they proposed trap files for early detection so their system can have more time to analyze and determine if the software is good or bad. To avoid early detection methods, the attacker changed the order of encrypting files. According to our analysis, attackers began to change the method used to encrypt files, which not only effectively increased the speed of file encryption but also greatly reduced the detection rate of existing ransomware detection systems.

Rather than focusing on the detection and processing of the malware, some studies analyzed the files encrypted by ransomware [6] explain on recovering the original file from the backup system by detecting ransomware infected files. Content analysis by using complex events and consider the file lifecycle for combating ransomware is discussed [7]. This paper starts from the observation of the status of the file itself and uses a variety of different formats of normal files and infected files. The files infected by the ransomware are mixed as a data set, and the machine learning model is trained through SVM to detect whether the files are maliciously encrypted so that the system has sufficient time to back up or block the service, and remediate the user’s files to minimize the damage. We propose a new method of fast screening of documents, the accuracy of the model is as high as 0.92.

II. RELATED WORK

In the last decade, machine learning techniques are widely used in ransomware detection and prediction. Cusack et al. propose a technique of extracting features from network traffic data and using random forest binary classifier to build a model. This model can reach a detection rate of 86% [8]. Takeuchi et al. have proposed ransomware detection by making API calls into consideration. They ran real ransomware samples in a sandbox environment and obtain important specific API call information to create a feature database and use the support vector machine to classify the samples [9]. Alhawi et al. have done an analysis of network traffic for windows ransomware. They focused on a conversational-based analysis and used J48, Decision tree classifier to calculate detection accuracy [10]. Subash et al. developed a reverse engineering framework incorporating feature generation engines and machine learning (ML) to efficiently detect ransomware. The Experimental results show the detection accuracy of ransomware samples which varied from 76% to 97% based on the ML technique used [11]. Shaukat et al. presented a layered defense system based on analysis of an extensive dataset of Ransomware families. It combined several layers include static and dynamic analysis layers to generate a novel compact set of features, a strong trap layer in early detection, machine learning for unearthing zero-day intrusions, and a file backup layer for encrypted files backup [12]. Adamu and Awan extracted five attributes from 30,000 attributes feature selection. Then apply support vector machine algorithm of which RMSE of 0.179 was obtained and classifying ransomware with 88.2% accuracy [13]. Lee et al. used the different characteristics of Entropy before and after file encryption, various types of entropy are regarded as input feature values of machine learning methods to identify whether the file has been encrypted [6]. Hirano and Kobayashi created and evaluated the machine learning model by using Random Forest, Support Vector Machine, and K-Nearest Neighbors and used five features of storage access patterns. They have achieved F-measure rate of 98% [14]. Maniath et al. proposed API call sequence of a process can be used as a metric to identify the behavior of a process. They developed a model to classify a given binary as ransomware or benign [15].

Indeed, the detected models proposed above do have good detection rate. However, due to the maturity of the hacking techniques, many tool sets, including malware are easily available. In addition, the biggest problem is that the hacker can use a variety of technologies such as code Obfuscation, packing and communication encryption [16]–[19] to change or hide the original features to avoid detection and reduce the detection rate. Therefore, we hope to find the key point that are difficult for hackers to change or the main process in the whole attack process as the features of our detection model to extend the validity time of our detection model. Most research focuses on analyzing the malware itself, including the static structure of the malware itself, the API called after dynamic execution and its network communication behavior, but these are more likely to be circumvented by hackers using existing technologies. According to our understanding of the running process of ransomware, its ultimate goal is to encrypt files on the hacked computer. This step is inevitable. Therefore, we hope to detect ransomware by quickly detecting whether the file is encrypted.

According to our research, entropy is used as a method to detect encrypted files [20], [21]. So we use it as one of the important features of our model. The EBDM proposed by Lee et al. [6], use the entropy as the feature and has a good detection rate. However, our recent research found that some new ransomware began to change the way of encrypting files, changing the complete file encryption to interlaced content.
TABLE 1. File types.

| .7Z | .DLL | .EXE | .HLP | .JPG |
|-----|------|------|------|------|
| .SYS | .BAT | .DLL | .DOC | .DOCX |
| .ODT | .PDF | .PPT | .PPTX | .RAR |
| .EXE | .GIF | .TMP | .TXT | .XLS |
| .XLSX | .ZIP |

encryption to increase the encryption speed, resulting in a decrease in the detection rate of the EBDM. Therefore, in this study we propose with more effective features to improve the detection rate.

III. PROPOSED METHODOLOGY
In this study, we found that there is a study that proposed an effective analysis system for identifying ransomware, but the detection rate is not good enough in some specific ransomware. In order to improve the detection rate of crypto ransomware that uses different methods to encrypt files, we first practically analyze the encrypted file format, then test the best entropy metric as feature values and use the SVM model to deploy. The reason we use SVM is its advantages in performance and detection rate in malware detection [22]. Therefore, we will directly adopt this model as our method and focus on the optimization research of better features extraction to identify different encrypted files.

A. DATASET DESCRIPTION
We collected 4 kinds of crypto ransomware, including WannaCry, Phobos, GandCrab and Globelmposter which will create the encrypted files with two different formats. Then we use 22 kinds of normal file formats (shown in Table 1) in the windows operation system as the targets and use the ransomware to create 1000 encrypted files as dataset for model training.

To keep the model from befuddling typical encrypted documents and malevolent encrypted files, we have specially added a variety of normally encrypted file formats such as rar, 7z, and zip in the dataset to reduce the false positive rate of the model.

To keep the model from befuddling typical encrypted documents and malevolent encrypted files, we have specially added a variety of normally encrypted file formats such as rar, 7z, and zip in the dataset to reduce the false positive rate of the model.

B. FEATURES EXTRACTION
Features extraction is one of the most important aspects in Machine Learning. As we mentioned above, May and Laron [7] proposed an Entropy Based Detection Method (EBDM for short) which is one useful method of evaluating the performance of cipher text generated from cryptography. They used three different types of entropy measures: Shannon entropy (1), the most common entropy (2), Rényi entropy with $\alpha = 2$ (collision entropy) (3) and compression estimate. In addition, they also use the file type as features, and have a very good detection rate. However, we found that four detecting specific ransomware (such as Globelmiposter), its performance is not ideal as shown in Fig. 1.

In order to understand the differences of various encrypted files in detail and find out their characteristic value. We use a virtual machine to build an analysis environment, and then use the four kinds of ransomware we collected to encrypt and analyze different types of files. We found that the file format encrypted by WannaCry, Phobos and GandCrab is very different from the file format encrypted by Globelmiposter, as shown in Fig. 2 and Fig. 3. The four ransomware use AES128 or AES256 to encrypt the file, but the first three ransomware encrypt all the content of the file, while Globelmiposter encrypts the content at 16-byte intervals, and then fills in the computer information at the end of the file. We think the way Globelmiposter used could speed the process time and due to the encrypted file contains half information of the original file, so it reduce the entropy value and that is why it cause a low detection rate in Kyungroul Lee proposed method.

In order to improve the accuracy and detection rate, we added additional Rényi Entropy $\alpha = 5$, via different Entropy measurement methods, different nonlinear $H(p)$ values can be obtained, strengthening the difference between normal files and infected files. It took a lot of time to train, therefore we replaced compression estimate to Compression Ratio (4), we used Python’s Zlib library for compression test, after compressed, and we also got compression parameters. And when the encrypted file is converted into bits, the ratio...
C.-M. Hsu et al.: Enhancing File Entropy Analysis to Improve ML Detection Rate of Ransomware

FIGURE 2. Frequency of hexadecimal value of original and encrypted DOCX file.

FIGURE 3. 0: 1 ratio of the same file encrypted by ransomware.

of 0 to 1 will approach 0.5, therefore, it’s necessary to convert the bytes code of the file into 0 and 1 of 1 bit, calculating Shannon Entropy and Markov entropy based on Markov’s theorem (5), and look at the proportion of 0 in the file. Finally, our file features are as shown in Table 2.

### TABLE 2. File features list.

| Item | Features                          | Item | Features                          |
|------|-----------------------------------|------|-----------------------------------|
| 1    | Filename Extension                | 10   | Extremity Markov Estimate         |
| 2    | Extremity Shannon Entropy(bytes)  | 11   | Compression Ratio                 |
| 3    | Rényi Entropy($\alpha=2$ )       | 12   | Extremity Zero values account for file(bits) |
| 4    | Extremity Rényi Entropy($\alpha=5$ ) | 13   | Most Common Value Test Estimate |
| 5    | Shannon Entropy(bytes)            | 14   | Extremity Rényi Entropy($\alpha=2$ ) |
| 6    | Extremity Compression Ratio       | 15   | Markov Estimate                   |
| 7    | Shannon Entropy(bytes)            | 16   | Extremity Shannon Entropy(bits)   |
| 8    | Extremity Most Common Value Test Estimate | 17   | Zero values account for file(bits) |
| 9    | Rényi Entropy($\alpha=5$ )       |      |                                   |

C. FEATURES EXTRACTION

This research hopes to quickly calculate whether each file is infected or not, so we chose to use SVM as our classifier. SVM can convert our feature $x_i$ to the hyperplane through the $\varphi$ function, find two types of margins on the hyperplane, and maximize the margins (6). The points on the boundary are called Support Vectors. $C$ is used to regulate the penalty function when error occurs.

$$\min_{w,b,\xi} \left( \frac{1}{2} w^T w + C \sum_{i=1}^{l} \xi_i \right)$$
subject to $y_i(w^T \varphi(x_i) + b) \geq 1 - \xi_i, \xi_i \geq 0$ (6)

Support vector machines can also use Kernel Trick to transform features into a non-linear space, thus improving the classification effect. We used the POLY method of General Polynomial Time Decomposition Algorithms to improve the recognition rate, where $\gamma$, $r$, and $d$ are the Kernel parameters.

$$K(x_i, x_j) = (\gamma x_i^T x_j + r)^d, \gamma > 0$$ (7)

D. PROCESS

The identification process of this study is shown in Fig. 4. First, calculate the different Entropy of the file with 17 features, and input the features to SVM. Then the SVM will determine whether the files are infected by ransomware. If there is an infection, the output is 1, otherwise, the output is 0.

E. CROSS-VALIDATION

In order to ensure the validity of the recognition model, we use Holdout Cross Validation and K-fold Cross Validation to verify the recognition accuracy of our model. Holdout Cross Validation is randomly dividing data set into two
groups. One group is training data and the other is test data. We used 70% of the data as training data, and 30% as test data; K-fold Cross Validation divides the data set into K number of groups, each of the group in turn will be used as test data, with the rest of the group be used as training data, and the test will be run K times. After everything is finished, the average of the results and R-squared will be observed. In order to observe the stability of the model, we set K as 10.

IV. RESULTS AND DISCUSSION

In order to obtain more objective results in this study, we used Holdout Cross Validation to divide our data set into 70% for training and 30% for testing to obtain the following results. We trained SVM Linear and SVM kernel Trick (Poly) models separately to detect normal files and files encrypted by WannaCry, GandCrab, GlobeImposter, and Phobos. When we compared our results with EBDM, we had a total of 16 different models with all of our models’ detection rates are 82% and higher, with SVM kernel Trick (Poly) is more than 92%, showing that our new features can improve the detection rate, as shown in Fig. 5. Since Poly method shows a more accurate model than linear model, our future results will only show SVM (Poly) result.

Then we combined all 4 types of ransomware, with the method in EBDM and Proposed method, and we used SVM kernel trick (poly), we trained a model separately. There are 2 models in total, the result can be seen in Fig. 6. The detection rate of EBDM was only 52.71% for GlobeImposter attack, while the detection of the method of this paper is at least 85.17%. This result evidently shows that in the future, if we need to identify multiple types of ransomware attacks, you can use the features of this research as the input feature of the model to achieve a more accurate detection rate.

In order to ensure the accuracy of this research method, our model is analyzed by Precision(8), Recall(9) and F1-score (10), and when compared with EBDM. The results are shown in the following Table 3. Our method result surpasses EBDM result especially the Recall and F1 score of GlobeImposter. It can be observed that the EBDM couldn’t distinguish whether the file is infected or not, and the detection accuracy drops to 52.71%, which means that infected files couldn’t be accurately detected. It also proves that our method can be used for detection of semi-encryption type of ransomware like GlobeImposter and general full encryption type of ransomware.

\[
\text{Precision} = \frac{TP}{TP + FP} \quad (8)
\]

\[
\text{Recall} = \frac{TP}{TP + FN} \quad (9)
\]

\[
F1\_score = \frac{2TP}{2TP + FP + FN} \quad (10)
\]

In order to understand the stability of the model in this research and to determine whether the accuracy and other values in this research are objective, we used K-fold Cross Validation to test the model, and set K to 50 to obtain more sets of validation test. When standard deviation is higher, it means that the value of each index at each training and test set is unstable. The lower the standard deviation, the higher the stability of the model. In Table 4, the results show our standard deviation is quite small, which proves that our model is quite stable and can effectively distinguish whether files are encrypted by crypto ransomware.

All methods mentioned above are used to identify the encrypted files, and we compared the performance between...
It can be seen that in the future if detection of various ransomware types is needed, the SVM and features mentioned in this research as the input for the model to get a stable and accurate detection rate.

**REFERENCES**

[1] U. Urrooj, M. A. B. Maarof, and B. A. S. Rimy, “A proposed adaptive pre-encryption crypto-ransomware early detection model,” in *Proc. 3rd Int. Cyber Resilience Conf. (CRC)*, Jan. 2021, pp. 1–6.

[2] F. Noorbehbahani, F. Rasouli, and M. Saberi, “Analysis of machine learning techniques for ransomware detection,” in *Proc. 16th Int. ISC Conf. Inf. Secur. Cryptol. (ISCISC)*, Aug. 2019, pp. 128–133.

[3] N. Naik, P. Jenkins, J. Gillette, H. Mouratidis, K. Naik, and J. Song, “Lockout-tagout ransomware: A detection method for ransomware using fuzzy hashing and clustering,” in *Proc. IEEE Symp. Series Comput. Intell. (SSCI)*, Dec. 2019, pp. 641–648.

[4] B. Qin, Y. Wang, and C. Ma, “API call based ransomware dynamic detection approach using TextCNN,” in *Proc. Int. Conf. Big Data, Artif. Intell. Internet Things Eng. (ICBAIE)*, Jan. 2020, pp. 162–166.

[5] S. Sheen and A. Yadav, “Ransomware detection by mining API call usage,” in *Proc. Int. Conf. Adv. Comput. Commun. Inform. (ICACCI)*, Sep. 2018, pp. 983–987.

[6] K. Lee, S.-Y. Lee, and K. Yim, “Machine learning based file entropy analysis for ransomware detection in backup systems,” *IEEE Access*, vol. 7, pp. 110205–110215, 2019.

[7] M. J. May and E. Laron, “Combating ransomware using content analysis and complex file events,” in *Proc. 10th IFIP Int. Conf. New Technol., Mobility Secur. (NTMS)*, Jun. 2019, pp. 1–5.

[8] G. Cusack, O. Michel, and E. Keller, “Machine-learning-based detection of ransomware using SDN,” in *Proc. ACM Int. Workshop Secur. Softw. Defined Netw. Function Virtualization*, 2018, pp. 1–6.

[9] Y. Takeuchi, K. Sakai, and S. Fukumoto, “Detecting ransomware using support vector machines,” in *Proc. 47th Int. Conf. Parallel Process. Companion*, New York, NY, USA, Aug. 2018, pp. 1–6.

[10] O. M. Alhawi, J. Baldwin, and A. Dehghantanha, “Leveraging machine learning techniques for windows ransomware network traffic detection,” in *Cyber Threat Intelligence (Advances in information Security)*, Springer, 2018, pp. 93–106.

[11] S. Poudyal, K. P. Subedi, and D. Dasgupta, “A framework for analyzing ransomware using machine learning,” in *Proc. IEEE Symp. Secur. Comput. Intell. (SSCI)*, Nov. 2019, pp. 1692–1699.

[12] S. K. Shahtak and V. J. Ribeiro, “RansomWall: A layered defense system against cryptographic ransomware attacks using machine learning,” in *Proc. 10th Int. Conf. Commun. Syst. Netw. (COMSNETS)*, Jan. 2018, pp. 356–363.

[13] U. Adamu and I. Awan, “Ransomware prediction using supervised learning algorithms,” in *Proc. 7th Int. Conf. Future Internet Things Cloud (FiCloud)*, Aug. 2019, pp. 57–63.

[14] M. Hirano and R. Kobayashi, “Machine learning based ransomware detection using storage access patterns obtained from live-forensic hypervisor,” in *Proc. 6th Int. Conf. Internet Things, Syst. Manage. Secur. (IOTSMS)*, Oct. 2019, pp. 1–6.

[15] S. Maniath, A. Ashok, P. Poornachandran, S. A. U. Prem, and S. Jan, “Deep learning LSTM based ransomware detection,” in *Proc. Recent Develop. Control, Automat. Power Eng.*., Oct. 2017, pp. 442–446.

[16] S. B. Prapulla, S. J. Bhat, and G. Shobha, “Framework for detecting metamorphic malware based on opcode feature extraction,” in *Proc. 2nd Int. Conf. Comput. Syst. Inf. Technol. Sustain. Solution (CSITSS)*, Dec. 2017, pp. 1–4.

[17] S. Singh and G. Kaiser, “Metamorphic detection of repackaged malware,” in *Proc. IEEE/ACM 6th Int. Workshop Metamorphic Test.*, 2021, pp. 1–8.

[18] M. Talukder, S. Islam, and P. Falcarin, “Analysis of obfuscated code as the file size increases due to different compression rate calculation strategies mentioned in part 3. Therefore, during the pre-encryption period, the faster the identification of infected files allows the system to have more time to block service or back up user’s files.

**V. CONCLUSION**

This research aims to propose a new strategy for real-time repair of computers that have been already attacked by ransomware, utilizing the file entropy of the system as the main identification object of the machine learning model, and propose a more efficient method to watch whether files are encrypted. We proposed to use SVM as a classifier, by the means of the file types, multiple entropy, file compression ratio as well as 0:1 scale and extracted a total of 17 features from 4 encrypted ransomware files in order to establish a detection model. In addition, we used Holdout Cross validation and K-fold Cross Validation to ensure our model’s validity. The result shows that when tested against various ransomware with different encryption methods, the method of this research can maintain the detection rate of 85.17% (where the detection rate of SVM kernel Trick (Poly) exceeds 92%). It can be seen that in the future if detection of various ransomware types is needed, use the SVM and features
CHIA-MING HSU received the bachelor’s degree from the National Defense University Chung Cheng Institute of Technology, in 2000, and the master’s degree from the Department of Information Management, National Defense University, China, in 2007, with a thesis on the topic of IP differential service queuing scheme based on time token. He is currently pursuing the Ph.D. degree with the Department of Electronic and Computer Engineering, National Taiwan University of Science and Technology. He is currently working as a Cyber-Threat Analyst in the government agency. His current interests include cyber security, especially on APT threat, malware analysis, and machine learning.

CHIA-CHENG YANG received the bachelor’s degree from the Department of Electrical Engineering, Feng Chia University, in 2019. He is currently pursuing the master’s degree with the Department of Electronic and Computer Engineering, National Taiwan University of Science and Technology. His current interests include PPG signal analysis, data analysis, and machine learning.

HAN-HSUAN CHENG received the bachelor’s degree from the Department of Electronic Engineering, National Taiwan University of Science and Technology, in 2020, where he is currently pursuing the master’s degree with the Department of Electronic and Computer Engineering. His current interests include malware reverse engineering, machine learning, and 5G signal analysis.

PAUL E. SETIASABDA received the bachelor’s degree in electrical engineering from Universitas Pelita Harapan, Indonesia, with a research in utilizing software defined-radio as GSM networks. He is currently pursuing the master’s degree in electronic and computer engineering from the National Taiwan University of Science and Technology. In 2015, he had an internship opportunity at the biggest telecommunication company in Indonesia, PT. Telekomunikasi Indonesia (Telkom) as the Fiber Optics Specialist and Maintenance. From 2017 to 2018, he also worked for one and half years at Keiretsu Forum Indonesia, a forum that gathers a lot of startups for investment and mentoring. He had the chance to work with a lot of technological startups that was growing during that time. He is currently doing an internship at an e-commerce company Mooimom as the IT Specialist.

JENQ-SHIOU LEU (Senior Member, IEEE) received the B.S. degree in mathematics and the M.S. degree in computer science and information engineering from the National Taiwan University, Taipei, Taiwan, in 1991 and 1993, respectively, and the Ph.D. degree (on a part-time basis) in computer science from the National Tsing Hua University, Hsinchu, Taiwan, in 2006. He was with Rising Star Technology, Taiwan, as an Research and Development Engineer, from 1995 to 1997, and worked in the telecommunication industry (Mobital Communications and Taiwan Mobile), from 1997 to 2007, as an Assistant Manager. In February 2007, he joined the Department of Electronic and Computer Engineering, National Taiwan University of Science and Technology, as an Assistant Professor. From February 2011 to January 2014, he was an Associate Professor. From February 2014 to July 2017, he was a Professor and the Vice Chairperson. Since August 2017, he has been a Professor and the Chairperson. His research interests include heterogeneous network integration, mobile service and platform design, distributed computing (P2P and cloud computing), and green and orange technology integration. He has published extensively in these areas, with 58 SCI-indexed journal articles, 57 conference papers or book chapters, and led 12 MOST project, 12 industry-academia projects, and two cross-university projects in the past ten years.