Attention-Based Multidimensional Deep Learning Approach for Cross-Architecture IoMT Malware Detection and Classification in Healthcare Cyber-Physical Systems

Vinayakumar Ravi, Tuan D. Pham, *Senior Member, IEEE*, and Mamoun Alazab, *Senior Member, IEEE*

***Abstract*— A literature survey shows that the number of malware attacks is gradually growing over the years due to the growing trend of Internet of Medical Things (IoMT) devices. To detect and classify malware attacks, automated malware detection and classification is an essential subsystem in healthcare cyber-physical systems. This work proposes an attention-based multidimensional deep learning (DL) approach for a cross-architecture IoMT malware detection and classification system based on byte sequences extracted from Executable and Linkable Format (ELF; formerly named Extensible Linking Format) files. The DL approach automates the feature design and extraction process from unstructured byte sequences. In addition, the proposed approach facilitates the detection of the central processing unit (CPU) architecture of the ELF file. A detailed experimental analysis and its evaluation are shown on the IoMT cross-architecture benchmark dataset. In all the experiments, the proposed method showed better performance compared with those obtained from several existing methods with an accuracy of 95% for IoMT malware detection, 94% for IoMT malware classification, and 95% for CPU architectures classification. The proposed method also suggests a similar performance with an accuracy of 94% on the Microsoft malware dataset. Experimental results on two malware datasets indicate that the proposed method is robust and generalizable in cross-architecture IoMT malware detection, classification, and CPU architectures classification in healthcare cyber-physical systems.**

***Index Terms*— Cybercrime, cybersecurity, deep learning (DL), federated learning, healthcare, Internet of Medical Things (IoMT), malware.**

I. Introduction

THE healthcare system has been one of the important sectors in developing countries with the ongoing increase in revenue and work opportunities. Earlier times, the only way to identify unusual conditions was to conduct a thorough clinical examination on the facilities of the hospitals. Internet-of-Things (IoT) is a network of interconnected, internet-connected entities that can gather and transmit information without the need for human interaction across a wireless connection [1]. IoT gives humans more freedom to interact and work with things. Remote health monitoring is an effective means of providing good preventive care and early treatment to high-risk individuals. Because of substantial advances in IoT principles, these monitoring devices are becoming more practical [2]. Limitations in access to medical resources, the growing senior population with serious illnesses and their need for remote patient monitoring, rising medical expenses, and the requirement for e-health make the IoT an intriguing topic in healthcare.

The Internet of Medical Things (IoMT) is a cutting-edge way of linking medical supplies and related software to healthcare systems through internet protocols [3]. The adoption of IoMT as a patient monitoring system allows for real-time monitoring employing smart wearable monitoring technologies. IoMT can assist us in identifying potential abnormalities in our body. Through the cognitive scope of the IoT, an effective IoMT tool would monitor patient health conditions in real time, foresee, and alarm against medical emergencies and support improved health. Patients' health records must be transferred to remote devices for assessment due to the inadequate data storage and computational resources of local IoT systems, which might possibly result in information leakage of patient data as well as the network's exposure to possible attacks [4]. Due to the massive fast development and deployment of IoT applications, privacy issues in IoT systems are only continuing to get worse. This brings up the chances of exploiting the online services to execute various sorts of intrusions in the IoT platform. In the context of IoMT, which focuses on the communication and control of smart medical equipment, it is a very critical concern [5]. In this setting, cyberattacks, such as replay, man-in-the-middle, impersonation, password guessing, and denial of service (DoS), are all achievable. Attackers will employ malware to exploit IoMT systems to get unauthorized entry to them and manipulate them virtually. Hackers implement networks of attacker systems with botnets, such as Mirai, Reaper, Echobot, Emotet, Gamut, and Necurs, to spread malware in IoMT networks [6].

Researchers in the field of cybersecurity employ a variety of ways to defend the IoMT network environment from advanced threats. Data are examined and matched to an existing database in the signature-based approach, and if they appear in the database, they are identified as malware. The signatures of previously detected threats are employed to determine identical cyberattacks [7]. The anomaly detection technique uses statistical approaches to identify irregularities. Specification-based approaches rely on some kind of standard

Manuscript received 21 April 2022; revised 9 July 2022; accepted 8 August 2022. Date of publication 19 August 2022; date of current version 2 August 2023. *(Corresponding author: Vinayakumar Ravi.)*

Vinayakumar Ravi and Tuan D. Pham are with the Center for Artificial Intelligence, Prince Mohammad Bin Fahd University, Khobar 31952, Saudi Arabia (e-mail: vravi@pmu.edu.sa; tpham@pmu.edu.sa).

Mamoun Alazab is with the College of Engineering, IT and Environment, Charles Darwin University, Casuarina, NT 0820, Australia (e-mail: alazab.m@ieee.org).

Digital Object Identifier 10.1109/TCSS.2022.3198123

or rule set of what comprises appropriate behavior to decide whether or not a software is harmful. Access control is a technique that governs a user's or device's accessibility to the system's resources. Overall, all these methods are effective in detecting the existing attacks, and in addition, the anomaly based approach results in high false positive rate. Since most of the IoMT systems operate on Linux-based systems with different central processing unit (CPU) architectures, it has been a very challenging to researchers to develop an IoMT malware detection system that can be a platform-independent, robust, and more generalizable [8].

A recent literature survey shows that machine learning (ML) and its subset deep learning (DL) are employed nowadays for IoMT malware detection using Executable and Linkable Format (ELF) files [5]. In addition, DL approaches performance shown better in various applications compared with ML [4], [15]. Feature engineering is required in ML [16], and the cost involved in feature extraction from ELF files in malware analysis is high due to the need for domain experts to identify the right features, whereas DL is capable to take in the raw inputs and has the capability to achieve better performances compared with the ML. A Static analysis and a dynamic analysis are successfully employed approaches for feature extraction from ELF files [3]. The static features are operational code (Opcode), printable strings, control flow graphs (CFGs), ELF header statistics, and byte sequences, and dynamic features are memory, process information, and application programming interface (API) call traces or network traces. A feature extraction process in static and dynamic analyses requires forensic tools and virtual environment setup, respectively. In recent days, a hybrid of static and dynamic analysis approaches were considered with the aim to enhance the malware detection and classification rate. To overcome these limitations, this work considers the byte sequence of ELF files as an input to the DL model. The proposed approach is platform-independent, robust, and more generalizable for IoMT malware detection and classification in healthcare cyber-physical systems on a dataset containing four malware families from ten different CPU architectures. The major contributions of the proposed work are as follows.

1) An attention-based multidimensional DL approach for cross-architecture IoMT malware detection, IoMT malware classification, and CPU architectures classification in healthcare cyber-physical systems.
2) An embedding layer is integrated into the proposed model that transforms the bytes representation into a numerical form.
3) Architecture contains a multichannel convolutional neural network (CNN) and a bidirectional long short-term memory (LSTM) layer that helps to learn *n*-gram representation of spatial features and sequence of byte information.
4) The attention layer is used to select optimal features from the CNN and bidirectional LSTM layer.
5) Feature fusion of DL layers for IoMT malware detetion, IoMT malware classification, and CPU architectures classification.
6) Detailed experiments and evaluation of the models shown for IoMT malware detection, IoMT malware classification, and CPU architectures identification using byte sequences of two benchmark big datasets.
7) A t-distributed stochastic neighbor embedding (t-SNE) visualization approach was employed to ensure that the learned features were meaningful for IoMT malware detection, IoMT malware family classification, and CPU architectures identification in healthcare cyber-physical systems.

The rest of this article is organized as follows. Section II includes a literature survey on malware analysis, the proposed architecture for malware analysis in Section III, information of malware datasets in Section IV, information of statistical metrics in Section V, and Section VI includes experiments, results, and its discussions, and finally, the conclusion and future works are placed in Section VII.

II. Literature Survey

There is a detailed analysis of several malware detection methods in the IoT/IoMT communication environment as well as a comparison to evaluate the effectiveness of various schemes, which will help in identifying the advantages and challenges of the various methods [17]. A literature survey on healthcare security shows that there are limited studies for IoMT malware detection and its classification using ML and DL [3], [5]. A detailed study on life cycle of IoT malware is shown by Alrawi *et al.* Also, the study examined how malware works and how code flows before it is detected, utilizing static, dynamic, hybrid, and memory analyses [18]. In these studies, various features from static analysis and dynamic analysis are used as input to detect and classify the ELF file. Table I includes a summary of related important approaches for malware detection and classification in the IoT/IoMT communication environment in healthcare cyber-physical systems.

As an anomaly based detection approach, IoT-Keeper [19] monitors IoT networks for abnormal behavior and detects unauthorized online activity by incorporating fuzzy clustering techniques with fuzzy interpolation. IoT-Keeper recognizes suspicious behavior and immediately imposes strict restrictions on IoT devices. An evaluation approach for feature selection, CorrAUC [20], is presented, adopting a new feature selection technique based on the wrapper scheme. Then, using Shannon entropy, identified attributes were validated for malicious traffic detection in the IoT. A secure IoT-based with an ML-based fault-tolerant decision-making framework is proposed for the advanced healthcare system [21]. A hybrid of ML and cryptography-based approach is proposed for authentication and secure transmission of data in IoMT systems [22]. An explainable DL-based approach is proposed for IoMT advanced cyber threats detection [23].

An analysis of various ML algorithms performances is evaluated for intrusion detection in IoMT network. Binbusayyis *et al.* [24] have shown detailed experiments using the IoT-related dataset. An optimization-based feature engineering approach with deep neural network for intrusion detection is proposed for IoMT network [25]. The authors have claimed that the deep neural networks performed better than the ML. A cryptography-based hashing approach is proposed for IoMT network security and privacy (S&P) [26]. A blockchain-enabled framework is proposed for IoMT networks [27] using elliptic curve digital signature algorithm. A nonlinear neural network-based approach is proposed for cyber-attacks detection in industrial IoT networks [28]. A privacy preserving healthcare framework is proposed for IoT

TABLE I
SUMMARY OF MALWARE DETECTION AND CLASSIFICATION

Reference	Platform	Data type	Length of Byte	Objective	Approach	Dataset size	Classes	Accuracy
Nguyen et al. [9]	IoT	CFG	-	Detection	CNN	11200	2	98.7%
Alasmary et al. [10]	IoT	CFG	-	Detection	CNN	6000	2	99.7%
Alasmary et al. [10]	IoT	CFG	-	Classification	CNN	6000	3	99.7%
Dovom et al. [11]	IoT	Opcodes	-	Detection	Fuzzy pattern tree	1207	2	99.83%
Niu et al. [12]	IoT	Opcodes	-	Detection	XGBoost	4169	2	94.5%
Wan et al. [13]	IoT	Byte	128-1024	Detection	SVM	222K	2	99.9%
Wan et al. [13]	IoT	Byte	128-1024	Classification	SVM	222K	8	98.47%
Nghi Phu et al. [14]	IoT	CFG	-	Detection	SVM	5476 (MIPS)	2	99.19%
Nghi Phu et al. [14]	IoT	CFG	-	Detection	SVM	6560 (Intel 80386)	2	99.06%
Proposed (current work)	IoMT	Byte	2KB	Detection	Deep learning	36236	2	95%
Proposed (current work)	IoMT	Byte	2KB	Classification	Deep learning	36236	5	94%
Proposed (current work)	IoMT	Byte	2KB	Classification	Deep learning	36236	10	95%

networks [29]. To avoid privacy issues in data sharing in industrial IoT environment, a federated learning-based approach is proposed for malware detection [30]. A DL-based approach is proposed for healthcare malware detection in smartphone environment [31], and it employs an embedding layer that transforms bytes into numerical representation. Furthermore, the optimal features are extracted from embedding representation using DL, and malware detection is done using a fully connected network. A hybrid DL-based ensemble learning approach is employed for cross-architecture IoT malware detection. The authors have shown the performance of the proposed architecture on advanced RISC machine (ARM), Intel80386, Microprocessor without Interlocked Pipelined Stage (MIPS), and MIPS + Intel80386 with an accuracy of 99.98.

N-BaIoT [32] is a novel network-based intrusion detection methodology for IoT devices that employs deep autoencoders to identify suspicious online activity from exploited devices. Pudukotai Dinakarrao *et al.* [33] described a two-pronged approach to detecting malware throughout the runtime, in which a runtime malware detector (HaRM) is developed that uses hardware performance counter (HPC) values. To prevent malware spreading and degradation of network performance, these data are sent in real time into a stochastic model predictive controller. The research provides a dynamic analysis for IoT malware detection (DAIMD) [34] to help minimize IoT device damages by dynamically discovering both well-known IoT malware and emerging and variant IoT malware. Omni-layered cloud infrastructure is used to detect IoT malware with a CNN model.

Fuzzy-based IoT malware detection and classification study are shown by Dovom *et al.* [11] on edge computing. Though the analysis is shown on large-scale different datasets, the model relies on Opcode feature extraction. The CNN-based approach is proposed for IoT and Android malware detection using features from CFG [10]. These features are vulnerable to obfuscation. eXtreme gradient boosting (XGBoost) was employed on the Opcode feature for the detection of X86-based IoT malware [12]. In addition to Opcode, other feature sets, such as API and PE header, were considered in this study. Dynamic analysis features, such as memory, network, system calls, file system, and process, are considered for IoT malware detection [34]. The system can be bypassed by malware, which uses a limited virtual machine environment for its execution. The 1024 length of byte sequences of the entry point from various CPU architectures of ELF files is considered for IoT malware detection and classification and CPU architectures classification using support vector machine (SVM) [13]. The study resulted in less performance for malware classification, and it shows that more than 1024 length of byte sequences are required to achieve better performances for malware classification. CFG and printable string information are used as features from ELF files for IoT malware detection using CNN [9]. String information is used for the detection of IoT malware using DNN in Windows and Linux environments [8]. The study also discussed the importance of platform-independent IoT malware detection. The SVM-based approach is proposed for IoT malware detection using the CFG feature [14]. The extreme learning-based approach is studied in detail for IoT malware detection and classification and Ransomware detection using features of Opcode and system class [35]. In addition, the authors also claimed that the proposed approach can be deployed in safety-critical systems, such as electronic healthcare systems. Bidirectional LSTM with a spatial pyramid pooling approach is proposed for smart IoT malware detection using Opcode and API calls, and the experiments were included for obfuscated malware [36]. A hybrid of CNN-LSTM with a software-defined networking (SDN)-enabled approach is proposed for IoMT malware detection using Opcode [37]. The dataset used in this dataset is highly imbalanced, and this can be one of the reasons the proposed method reported a good IoMT malware detection rate. In addition, the size of the dataset is less, and overall, the dataset contains less than 1300 ELF files. SDN-based orchestration with a CNN-LSTM approach is proposed to detect and classify advanced cyber threats in IoMT [38]. A detailed study on various datasets is shown and their deployment in IoMT.

III. PROPOSED IoMT SYSTEM

The proposed architecture for IoMT malware detection, IoMT malware classification, and CPU architecture classification is shown in Fig. 1. The proposed framework employs a multidimensional DL approach to learn optimal feature representation that helps to accurately detect malware and classify them into malware categories based on byte representation of ELF files. Each dimension contains a sequence of byte sequence, byte representation, and ELF modules. These modules work together, and the detailed information of all the components of the proposed architecture is discussed in the following.

Byte Sequence: In the byte sequence layer, the byte information from ELF files is extracted. No preprocessing is required for the byte information except transforming all the byte sequences of ELF files to the same length. The entry point

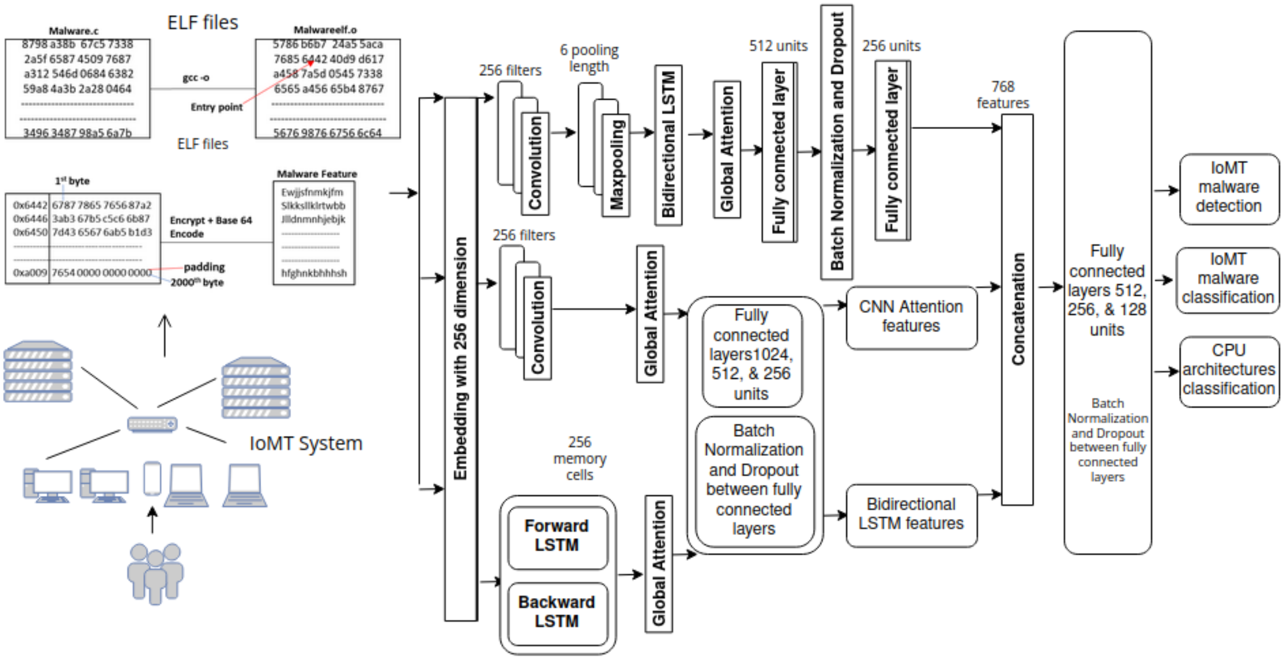

Fig. 1. Malware analysis system in IoMT.

address in the ELF header is identified, and then, the first 2k bytes from the entry point address are extracted. If the binary file length is less than 2k bytes, the byte sequence is padded with zero bytes for completion and maintains the consistent length among all the malware samples.

Byte Representation: The byte representation submodule leverages embedding concepts from natural language processing (NLP) to transform the byte information to numerical representation. Using tokenizer, the bytes are transformed into unigram representation, and all characters of bytes are in small letters of the English alphabet. Discriminating between small letters of character bytes and big letters of the character of bytes might require more complex architectures to learn optimal feature representation, and in some cases, it leads to a decrease in the model performance. Hence, in this work, all the characters of bytes are transformed into small characters of English alphabets. Since the data are not a natural language text, this work employs a simple embedding concept that includes a dictionary to assign a unique representation for each of the characters of bytes. A dictionary contains an unknown character that helps to assign a numerical representation if the model sees any character of bytes that is not present in the dictionary. Using a dictionary, the character of bytes data is replaced by an index of a dictionary. The Keras embedding layer takes three parameters as input, and they are as follows: 1) number of unique characters in bytes data; 2) size of the character embedding representation; and 3) size of the input sequence The number of unique characters present in the ELF files dataset is 69, and we set the size of the dictionary to 70 with an additional character for an unknown character. The embedding dimension is set to 256, and the size of the input sequence is 2000. For each character, the embedding layer generates 256 dimension vector representation. An embedding layer transforms a discrete representation of a matrix into a low-dimensional continuous or dense matrix representation. Initially, the weights matrix of the embedding layer is initialized randomly, and furthermore, the weights are updated using a network optimizer during backpropagation.

ELF Files Classification: Furthermore, numerical sequence representation of bytes follows DL layers, such as CNN, CNN with attention, and bidirectional LSTM with attention. The CNN and bidirectional LSTM layer help to learn *n*-gram representation of spatial features of bytes values and sequence information of byte information, respectively. Both CNN and bidirectional LSTM layer takes input from an embedding layer and employs CNN and bidirectional LSTM, respectively.

CNN is a well-known approach for image classification, and in the case of text data, CNN employs a 1-D convolution kernel operation that facilitates learning spatial patterns by a sliding window on bytes representation of embedding vectors. In convolution, more than one filter is used to learn multiple features with varying window sizes. The filter operation is applied for each possible window of characters in a byte sequence to generate a feature map. In CNN, the number of filters is set to 256 with a filter length of 6 and activation function to rectified linear unit (ReLU). The CNN layer follows a max-pooling layer that acts as a downsampling operation to extract maximum value from a feature map. Overall, CNN and pooling employed layers in the proposed approach are placed in a hierarchical fashion to learn a smaller number of high-level features from lower level features that are effective for classification. The main idea of the max-pooling operation is to capture the most important feature. Max-pooling is set to 6 in this work. Next, two fully connected layers were included with batch normalization and dropout between the fully connected layers with neurons 512 and 256.

Bidirectional LSTM is an extension of the LSTM layer that learns both the past and future information of character bytes in a byte sequence data bypassing the byte character sequence data into forward and backward directions inside a hidden layer during backpropagation. It contains 256 memory cells, tanh activation function, sigmoid recurrent activation function, uniform distribution initializer, and orthogonal recurrent initializer. At each time step, the hidden state is estimated by concatenating the forward and backward hidden states.

Since the byte sequence of the ELF files is too long, there may be a possibility that the CNN and bidirectional LSTM model capability might decrease in identifying the right features for IoMT malware detection, IoMT malware classification, and CPU architecture classification. To overcome this, an attention layer is included to CNN after the first convolution layer and bidirectional LSTM that helps to learn important information from a long byte sequence from ELF files. In addition to excluding irrelevant information from byte sequence, the attention approach reduces the computational complexity of the model. Instead of sending the output of the last bidirectional LSTM cell in the bidirectional LSTM model and output convolution layer in the CNN model, in this work, the whole sequence information of all the hidden states is passed into a global attention layer. In a global attention layer, global attention is employed. Global attention is also called additive attention, which is similar to soft attention [39]. In the global attention layer, the learnable function is formed by performing tanh operation on the hidden sequence vectors, such as features of bidirectional LSTM or CNN. Next, the informative bidirectional LSTM or CNN feature is extracted by applying softmax on the learnable function. Finally, new feature representation is estimated by applying a weighted average of the informative bidirectional LSTM or CNN features and hidden sequence vectors obtained by LSTM or CNN. The attention layer follows three fully connected layers in a bidirectional LSTM and CNN model with neurons 1024, 512, and 256. Between the fully connected layers, dropout and batch normalization are included. With the aim to learn more optimal feature representation, the attention features of the bidirectional LSTM and CNN model passed into more than one fully connected layer. This helps to map the features into higher dimensional space in which more optimal byte feature representation is extracted. To avoid overfitting and increase the speed during training, dropout and batch normalization are included. During forward propagation, dropout randomly removes a few hidden neurons and their connections. During training, batch normalization stabilizes the whole learning process by applying a transformation that maintains the mean output close to 0 and the output standard deviation close to 1.

The last fully connected layer feature dimensions of CNN, CNN with attention, and bidirectional LSTM with attention are $n \times 256$, $n \times 256$, and $n \times 256$, respectively, where n denotes the number of ELF files. The last fully connected layer in CNN, CNN with attention, and bidirectional LSTM with the attention model is fused using a concatenate layer in Keras and later passed the fused features into more than one fully connected layer for classification. The first fully connected layer contains 512 neurons, the second fully connected layer contains 256 neurons, and the third fully connected layer contains 128 neurons. The final classification layer classifies the ELF files into either malware or legitimate in the case of IoMT malware detection using one unit with a sigmoid activation function. Softmax with the number of classes as a value is used in units in the case of IoMT malware classification and CPU architecture classification. The weights are fine-tuned during training the models by considering the binary cross entropy as a loss function in the case of IoMT malware detection and categorical cross entropy as a loss function in the case of IoMT malware classification and CPU architecture classification.

TABLE II
CDMC-2020-IoMT-MALWARE DATASET

IoMT Malware Family	No. of Samples	Training	Testing
Android	247	200	47
Bashlite	7091	5448	1643
Benign	19975	15393	4582
Mirai	8418	6472	1946
Tsunami	505	388	117
Total	36236	27901	8335

IV. DESCRIPTION OF MALWARE DATASETS

In this work, cybersecurity data mining competition (CDMC)-2020-IoMT-Malware [40] and Microsoft malware classification: Big-2015 [41] datasets are used to evaluate the performances of the proposed method. The detailed statistics of the datasets are included in the Tables II–IV.

The ELF format of benign and malicious Linux programs was collected from various sources. Both benign and malicious Linux programs contain ten different CPU architectures. More details and statistics of CPU information are shown in Table III. The entry point address in the ELF header is identified, and then, the first 2k bytes from the entry point address are extracted. If the binary file length is less than 2k bytes, the byte sequence is padded with zero bytes for completion and maintains the consistent length among all the malware samples. The extracted 2k byte sequences in American Standard Code for Information Interchange (ASCII) format are encoded with the encryption cipher to scramble the original ELF program content bytes. The encrypted content is fed to the base64 encoder to obtain Radix64 human-readable format data representation, which is added in the third column of the binary sample in the dataset.

Microsoft Malware Classification: Big-2015 dataset was used in the malware classification challenge in 2015. The dataset provides assembly source code (ASM) and bytes files for nine different malware families. In this work, only the bytes dataset was used, and preprocessing was employed to remove the unnecessary information from the bytes file, i.e., starting address. The nine different malware family categories cover malicious aspects, such as victim credential stealing, exfiltrating user browsing data, sending spam e-mails, rogue anti-spyware advertisements on websites, and backdoors for remote access. The information of malware families and their statistics are included in Table IV.

Using Sklearn train-test-split approach, the datasets of CDMC-2020-IoMT-Malware and Microsoft malware classification: Big-2015 are divided into train, valid, and test. These datasets are unique and disjoint to each other.

V. STATISTICAL METRICS

Accuracy is the classifier's ability to classify all legitimate ELF files as legitimate ELF files and all malware ELF files as malware ELF files

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}. \quad (1)$$

Precision is a measure of the classifier's ability to not mark malware ELF files as legitimate ELF files

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}. \quad (2)$$

TABLE III
CPU INFORMATION IN CDMC-2020-IOMT-MALWARE

CPU	No. of Samples	Training	Testing
armel	8994	6927	2067
mips64eb	1040	798	242
mipseb	9052	6949	2103
mipsel	4976	3861	1115
ppceb	2345	1849	496
sh4el	1276	970	306
sparceb	1215	932	283
unknown	1003	788	215
x86_64el	2740	2070	670
x86el	3595	2757	838
Total	36236	27901	8335

TABLE IV
BIG-2015 DATASET

Malware Family	No. of Samples	Training	Testing
Ramnit	1541	1116	425
Lollipop	2478	1757	721
Kelihos_ver3	2942	2049	893
Vundo	475	340	135
Simda	42	32	10
Tracur	751	527	224
Kelihos_ver1	398	279	119
Obfuscator.ACY	1228	884	344
Gatak	1013	732	281
Total	10868	7716	3152

Recall is a measure of the classifier's ability to mark all legitimate ELF files as legitimate ELF files

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \tag{3}$$

$F1$-score is the weighted average of precision and recall

$$\text{F1score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4}$$

where TP, FP, TN, and FN are true positive, false positive, true negative, and false negative, respectively. The accuracy, precision, recall, and $F1$-score are estimated from the values obtained from the confusion matrix. The confusion matrix is a matrix that displays and compares actual values with the predicted values. The dimension of the confusion matrix depends on the number of classes in the dataset, for example, in the case of malware detection, i.e., classifying the ELF file as legitimate or malware, the dimension of the confusion matrix is 2 × 2. Since the datasets used in this work are not balanced, precision, recall, and $F1$-score reported in this work are macro, not weighted. Macro-metric computes the precision, recall, and $F1$-score for each class and returns the average without considering the proportion for each class in the IoMT dataset.

VI. EXPERIMENTS, RESULTS, AND DISCUSSION

All the models were implemented using TensorFlow as back end with Keras front-end library, and scikit-learn was used for implementing ML algorithms. The experiments for all the models were run on Kaggle NVidia K80 GPUs.

The proposed architecture in this work is based on a hybrid of CNN and bidirectional LSTM with attention and embedding layers. Layers of DL have many parameters, and choosing the optimal parameter is important to achieve better performance for IoMT malware detection, IoMT malware classification, and CPU architecture classification. In the beginning, the moderate size of DL architecture was employed. The network contains an input layer, hidden layers, and an output layer. An output layer contains M neurons in IoMT detection and N neurons in IoMT malware classification and L neurons for CPU architecture classification. Here, M, N, and L denote the number of classes. We run several trials of experiments to identify the best parameters for network parameters and network architecture for IoMT malware detection, IoMT malware classification, and CPU architectures classification. Training accuracy and loss curves are shown in Figs. 2 and 3, respectively. The figure shows the increase in accuracy and the decrease in loss by the proposed approach across 100 epochs in IoMT malware detection, IoMT malware classification, and CPU architecture classification. To identify the optimal parameters for the optimizer, learning rate, and batch size, various trials of experiments were run, and the parameters of the optimizer, learning rate, and batch size were set to adam, 0.0001, and 64. Experiments with adam performed better than stochastic gradient descent (SGD), and the learning rate with 0.0001 was better compared with other learning rates, such as 0.001, 0.01, and 0.1. Experiments were run with the batch sizes of 16, 32, 64, 128, and 256. The proposed model with batch size 64 performance was better compared with 16 and 32, and there was no performance improvement after the batch size of 64.

The detailed results for IoMT malware detection are shown in Table V. The proposed approach performed better than other DL approaches, such as attention-based recurrent neural network (RNN), LSTM, gated recurrent unit (GRU), CNN, and bidirectional LSTM with an accuracy of 95%. In addition to the accuracy statistical metric, the proposed approach showed better performances compared with the other DL approaches with a precision of 96%, a recall of 95%, and an $F1$-score of 95%. In detail, the proposed approach showed less misclassification rate for both malware and benign with misclassifying 320 samples of benign as malware and 57 samples of malware as benign. The confusion matrix of the proposed approach shows that the proposed approach has a good malware detection rate and a large number of misclassifications happened for benign samples. A detailed study can be done on this to understand the reason, and it helps to improve the current detection rate of malware. The main reason for misclassification can be due to the reason that there are no sufficient data samples for IoMT malware and benign. Data augmentation or generative adversarial-based approaches can be effectively employed on the existing IoMT datasets that help to generate sufficient data samples. The results of attention-based LSTM and GRU performed better than the attention-based RNN and CNN. The attention-based bidirectional LSTM showed similar performances as attention-based LSTM and GRU, but the misclassification rate for malware was less. This indicates that the bidirectional LSTM was effective compared with unidirectional LSTM. Attention-based CNN performed better than RNN with less misclassification rate for malware. Overall, the performance of RNN was less compared with the other DL models. With the aim to learn both spatial- and sequence-related optimal features from byte sequence, both CNN and LSTM models are combined with attention.

TABLE V
DETAILED RESULTS FOR IoMT MALWARE DETECTION AND CLASSIFICATION

Method	Accuracy	Precision	Recall	F1-Score	Confusion Matrix	Accuracy	Precision	Recall	F1-Score
	IoMT malware detection					IoMT malware classification			
A-RNN$_{\text{IoMT}}$	0.88	0.88	0.88	0.88	[3275 478] [486 4096]	0.87	0.84	0.72	0.74
A-LSTM$_{\text{IoMT}}$	0.92	0.92	0.92	0.92	[3435 318] [324 4258]	0.91	0.80	0.81	0.81
A-GRU$_{\text{IoMT}}$	0.92	0.92	0.92	0.92	[3473 280] [366 4216]	0.89	0.80	0.83	0.81
A-CNN$_{\text{IoMT}}$	0.90	0.91	0.90	0.90	[3260 493] [304 4278]	0.86	0.72	0.79	0.74
A-B-LSTM$_{\text{IoMT}}$	0.92	0.92	0.92	0.92	[3428 325] [308 4274]	0.91	0.84	0.84	0.84
A-M-CNN-B-LSTM$_{\text{IoMT}}$ (Proposed)	0.95	0.96	0.95	0.95	[3433 320] [57 4525]	0.94	0.96	0.87	0.91

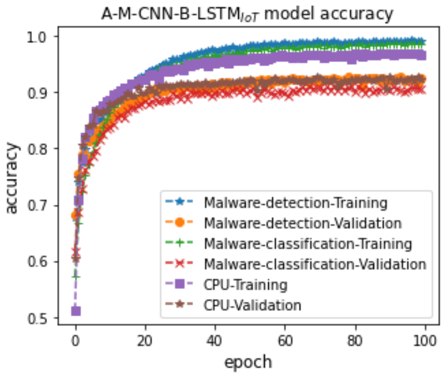

Fig. 2. IoMT model training accuracy.

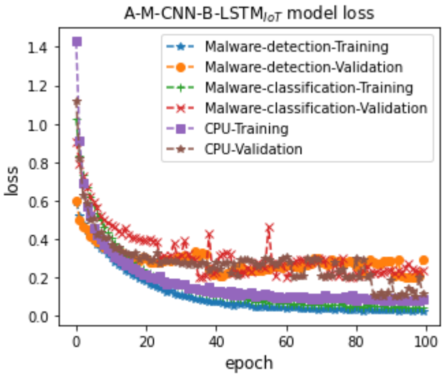

Fig. 3. IoMT model training loss.

Since the sequence length of the byte was long, bidirectional LSTM was considered instead of unidirectional LSTM, and this helped to achieve a better IOMT detection rate compared with other methods.

IoMT malware classification results are included in Table V. The proposed model performed a better accuracy of 94% compared with other DL models, such as attention-based RNN, LSTM, and CNN. Precision, recall, and $F1$-score of the proposed approach for IoMT malware classification are 86%, 87%, and 91%, respectively. The proposed approach for IoMT malware classification was better in terms of accuracy, precision, recall, and $F1$-score. Attention with LSTM and GRU models performed better than the RNN and CNN. The results shown by attention-based CNN and RNN are almost the same with a high misclassification rate compared with other models. The main reason for less performance by RNN is because the longer byte sequences were not effectively handled by RNN over time sequences. However, this was handled by LSTM effectively and achieved better performances compared with RNN. Since the sequence length of the byte sequence is long, the performance shown by bidirectional LSTM was better compared with LSTM. Though CNN was effective for learning spatial-related feature representation over the character byte sequences, embedding sequence models of DL, such as LSTM, can show better performances for IoMT malware classification. Thus, the proposed approach combines the CNN and bidirectional LSTM with an attention model and achieved better performances compared with the other DL models for IoMT malware classification. Though the performance of the proposed model is good compared with the other DL models, the proposed approach has a higher misclassification rate. This is the main reason why the recall, $F1$-score, and precision of the proposed approach were less. The main reason is the dataset that is highly imbalanced, and the proposed approach is not based on cost-sensitive learning. During training the model, more importance can be given to the classes by assigning larger weights and lesser weights to the classes, which have a high number of data samples. This type of training process avoids biased training, and most importantly, the proposed method performances can be improved. This will be considered as one of the future work directions of the proposed work.

Along with the IoMT malware detection and IoMT malware classification, the performance of the proposed approach based on attention with a hybrid of CNN and bidirectional LSTM is shown for CPU architecture classification. A detailed result for CPU architecture classification is included in Table VI. This was done due to the recent IoMT applications using heterogeneous CPU architectures [3], [5]. Identifying the CPU architecture of the malware and benignware applications of IoMT helps to know detailed information of the IoMT application, and this can be used as a feature in developing IoMT malware detection and IoMT malware classification. The proposed approach performed better than attention-based RNN, LSTM, CNN, and bidirectional LSTM models with an accuracy of 95%, a precision of 96%, a recall of 93%, and

TABLE VI

Results for CPU Architectures Classification in IoMT Malware and Malware Family Classification Using Big-2015

Method	Accuracy	Precision	Recall	F1-Score	Accuracy	Precision	Recall	F1-Score
	IoMT CPU architecture classification				Microsoft malware classification (Big-2015)			
A-RNN$_{\text{IoMT}}$	0.90	0.89	0.84	0.85	0.90	0.89	0.84	0.85
A-LSTM$_{\text{IoMT}}$	0.92	0.93	0.90	0.91	0.92	0.93	0.90	0.91
A-GRU$_{\text{IoMT}}$	0.93	0.93	0.90	0.91	0.93	0.93	0.90	0.91
A-CNN$_{\text{IoMT}}$	0.91	0.91	0.87	0.88	0.91	0.91	0.87	0.88
A-B-LSTM$_{\text{IoMT}}$	0.93	0.93	0.90	0.92	0.93	0.93	0.90	0.92
A-M-CNN-B-LSTM$_{\text{IoMT}}$ (Proposed)	0.95	0.96	0.93	0.94	0.95	0.96	0.93	0.94

an $F1$-score of 94%. The models based on LSTM, GRU, and CNN with attention performed better than the RNN for CPU architectures classification. Also, unidirectional LSTM and GRU model performances were less compared with bidirectional LSTM for CPU architecture classification. Though the performance of the proposed method is considered to be good, still some CPU architectures have large misclassification, and this can be avoided to enhance the performance of the proposed method for IoMT malware detection and IoMT malware classification.

Robustness and generalization are very important in DL. To show that the proposed method is robust and generalizable, the proposed method performance is evaluated on additional datasets, such as the Microsoft malware Big-2015 dataset, and its results are included in Table VI. The proposed approach has shown similar performance for malware classification with an accuracy of 94%, a precision of 90%, a recall of 89%, and an $F1$-score of 89%. Most importantly, the proposed approach performed better than the attention-based RNN and CNN. Also, LSTM and CNN performed better than RNN, and unidirectional LSTM performance was less compared with bidirectional LSTM. This indicates that the proposed method is robust and generalizable on unseen malware samples detection and malware classification. However, the proposed model for IoMT malware detection, IoMT malware classification, and IoMT CPU architecture classification may not be robust and generalizable on an adversarial modified dataset or in an adversarial environment. The detailed case study of the proposed method performance in an adversarial environment is not shown in this work, and this is important future work directions of the proposed work. Various available adversarial attacks in the literature can be effectively utilized to evaluate the proposed work in an adversarial environment. Otherwise, the application of generative adversarial network (GAN)-based approaches can be employed to generate similar datasets from different distributions, and the performance of the proposed approach can be evaluated on the GAN-based generated malware.

Recent literature survey shows that DL model interpretation and its explainability are considered to be important, which helps to find out the reason behind the better performances for IoMT malware detection, IoMT malware classification, and CPU architectures classification in healthcare systems. This is mainly important, because the DL model extracts optimal features by passing data into more than one layer and understanding how the model that learns optimal features over the hidden layers is important. t-SNE feature representation can be effectively used to represent the hidden layer features. It is a simple method that takes $m \times n$ data and converts it into $m \times 2$ by employing a dimensionality reduction approach. These two dimensions are displayed in the X-axis and Y-axis of the figure, respectively. This kind of feature visualization permits checking the data samples from different classes in separate clusters. In this work, a principal component analysis (PCA) approach was employed for dimensionality reduction. In addition to the dimensionality reduction approach, t-SNE has several important parameters, and they are n_components, perplexity, learning rate, and iterations. In this work, n_components are set to 2, perplexity is set to 40.0, the learning rate is set to 150.0, and embedding initialization is set to PCA. The optimal parameters for t-SNE were set based on hyperparameter tuning. The t-SNE feature representation for IoMT malware detection, IoMT malware classification, and CPU architectures classification is shown in Fig. 4. The figure shows that still there is no clear separation between the Normal and Malware in IoMT malware classification. In IoMT malware classification, the clusters of benign, Mirai, and Bashlite are overlapped. Though the model has shown a clear separation between different CPU architectures, still most of them have some misclassification. t-SNE feature representation for Microsoft malware classification is shown in Fig. 4, and this also has overlapping clusters among malware families. Thus, further investigation and detailed study are required to avoid these misclassifications. Otherwise, the systems cannot be effectively deployed in healthcare systems, because even a single misclassification of malware can cause significant damages. Thus, the proposal of an improved model of this work can be considered as one of the significant directions toward future work. Mainly, the improved model has the capability to avoid misclassification, and the system can provide interpretation and explainability of the detection of malware to the end users.

The detailed experimental analysis and literature survey on IoMT malware detection show that the proposed method is considered to be better for malware detection in IoMT network in cyber-physical environment. The proposed method has the capability to classify the malware and identifying its CPU architectures along with the malware detection. The attention with multidimensional DL facilitates to extract important features by viewing the bytes data in multiple ways. Overall, the proposed method is considered to be optimal for IoMT malware detection, malware classification, and its CPU architectures detection. The data from various IoT devices need to be collected in the proposed work, and this type of data collections may not be always feasible in real-time IoMT networks. Thus, integrating federated learning approach to the proposed IoMT network can be considered as future work.

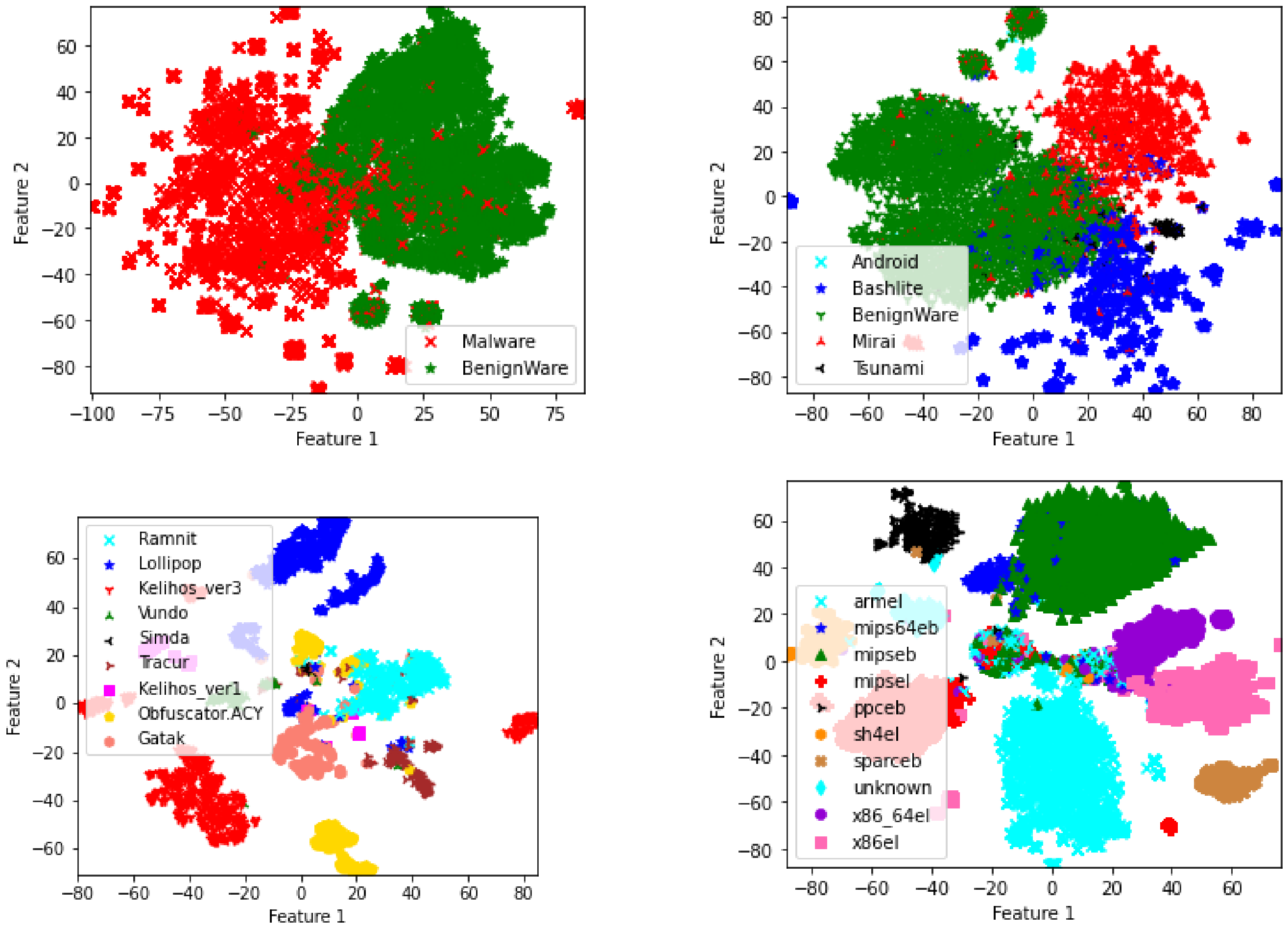

Fig. 4. t-SNE feature visualization for IoMT malware detection, IoMT malware classification, IoMT malware classification using Big-2015, and IoMT CPU architectures classification (left to right).

VII. Conclusion and Future Works

Automated malware detection and classification subsystems are an essential component in healthcare cyber-physical systems to overcome security threats and attacks. This work presents an attention-based multidimensional DL approach for cross-architecture IoMT malware detection and classification systems based on byte sequences extracted from ELF files. The characters in byte sequences are transformed into numeric representation using an embedding layer. An embedding layer has a sequence of DL layers, in between DL layers attention is introduced to focus on the important features of the byte sequence that are required to accurately detect malware and classify them into their malware categories. In addition to malware detection and malware classification, the proposed method can accurately identify the CPU architecture of the ELF file. In all the experiments, the proposed method outperformed other methods for malware detection, malware classification, and CPU architecture identification of ELF files. In particular, the proposed method showed 94% accuracy for IoMT malware detection, 95% for IoMT malware classification, and 95% for CPU architectures classification. Most importantly, it showed similar performances as the CDMC-2020-IoMT-Malware dataset on the Microsoft malware dataset for malware classification with an accuracy of 95%. This shows that the proposed method is robust and generalizable to unseen malware detection, malware classification, and CPU architecture identification. Since the proposed model is multichannel, additional features, such as Opcode, string, and so on, of ELF files can be included to improve the performances. This can be considered as future work. Also, recent literature shows that DL-based models are not robust against adversarial attacks. Thus, a detailed investigation and analysis of the proposed method have to be shown for malware detection, malware classification, and CPU architecture identification in healthcare cyber-physical systems in an adversarial environment. In addition, the proposed model performance has to be studied against a big dataset with more malware families.

References

[1] N. Gupta, J. Singh, S. K. Dhurandher, and Z. Han, “Contract theory based incentive design mechanism for opportunistic IoT networks,” *IEEE Internet Things J.*, early access, Aug. 31, 2021, doi: 10.1109/JIOT.2021.3109162.

[2] D. K. Sharma, K. K. Bhardwaj, S. Banyal, R. Gupta, N. Gupta, and L. Nkenyereye, “An opportunistic approach for cloud service-based IoT routing framework administering data, transaction, and identity security,” *IEEE Internet Things J.*, vol. 9, no. 4, pp. 2505–2512, Feb. 2022.

[3] R. U. Rasool, H. F. Ahmad, W. Rafique, A. Qayyum, and J. Qadir, “Security and privacy of internet of medical things: A contemporary review in the age of surveillance, botnets, and adversarial ML,” *J. Netw. Comput. Appl.*, vol. 201, May 2022, Art. no. 103332. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1084804522000017

[4] E. H. Houssein, M. Hassaballah, I. E. Ibrahim, D. S. AbdElminaam, and Y. M. Wazery, “An automatic arrhythmia classification model based on improved marine predators algorithm and convolutions neural networks,” *Expert Syst. Appl.*, vol. 187, Jan. 2022, Art. no. 115936. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417421012902

[5] M. Wazid, A. K. Das, J. J. P. C. Rodrigues, S. Shetty, and Y. Park, “IoMT malware detection approaches: Analysis and research challenges,” *IEEE Access*, vol. 7, pp. 182459–182476, 2019.

[6] S. Nandy, M. Adhikari, M. A. Khan, V. G. Menon, and S. Verma, “An intrusion detection mechanism for secured IoMT framework based on swarm-neural network,” *IEEE J. Biomed. Health Informat.*, vol. 26, no. 5, pp. 1969–1976, May 2022.

[7] M. Alhanahnah, Q. Lin, Q. Yan, N. Zhang, and Z. Chen, “Efficient signature generation for classifying cross-architecture IoT malware,” in *Proc. IEEE Conf. Commun. Netw. Secur. (CNS)*, May 2018, pp. 1–9.

[8] C. Hwang, J. Hwang, J. Kwak, and T. Lee, “Platform-independent malware analysis applicable to windows and Linux environments,” *Electronics*, vol. 9, no. 5, p. 793, May 2020.

[9] H.-T. Nguyen, Q.-D. Ngo, and V.-H. Le, “A novel graph-based approach for IoT botnet detection,” *Int. J. Inf. Secur.*, vol. 19, no. 5, pp. 567–577, Oct. 2020.

[10] H. Alasmary *et al.*, “Analyzing and detecting emerging Internet of Things malware: A graph-based approach,” *IEEE Internet Things J.*, vol. 6, no. 5, pp. 8977–8988, Oct. 2019.

[11] E. M. Dovom, A. Azmoodeh, A. Dehghantanha, D. E. Newton, R. M. Parizi, and H. Karimipour, “Fuzzy pattern tree for edge malware detection and categorization in IoT,” *J. Syst. Archit.*, vol. 97, pp. 1–7, Aug. 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1383762118305265

[12] W. Niu, X. Zhang, X. Du, T. Hu, X. Xie, and N. Guizani, “Detecting malware on X86-based IoT devices in autonomous driving,” *IEEE Wireless Commun.*, vol. 26, no. 4, pp. 80–87, Aug. 2019.

[13] T.-L. Wan *et al.*, “Efficient detection and classification of Internet-of-Things malware based on byte sequences from executable files,” *IEEE Open J. Comput. Soc.*, vol. 1, pp. 262–275, 2020.

[14] T. Nghi Phu, N. Dai Tho, L. Huy Hoang, N. Ngoc Toan, and N. Ngoc Binh, “An efficient algorithm to extract control flow-based features for IoT malware detection,” *Comput. J.*, vol. 64, no. 4, pp. 599–609, Apr. 2021.

[15] V. Acharya, V. Ravi, T. D. Pham, and C. Chakraborty, “Peripheral blood smear analysis using automated computer-aided diagnosis system to identify acute myeloid leukemia,” *IEEE Trans. Eng. Manag.*, early access, Aug. 27, 2021, doi: 10.1109/TEM.2021.3103549.

[16] H. Bhuyan, D. C. Chakraborty, S. Pani, and V. Ravi, “Feature and subfeature selection for classification using correlation coefficient and fuzzy model,” *IEEE Trans. Eng. Manag.*, early access, Apr. 19, 2021, doi: 10.1109/TEM.2021.3065699.

[17] A. Gatouillat, Y. Badr, B. Massot, and E. Sejdić, “Internet of medical things: A review of recent contributions dealing with cyber-physical systems in medicine,” *IEEE Internet Things J.*, vol. 5, no. 5, pp. 3810–3822, Oct. 2018.

[18] O. Alrawi *et al.*, “The circle of life: A large-scale study of the IoT malware lifecycle,” in *Proc. 30th USENIX Secur. Symp. (USENIX Secur.)*, 2021, pp. 3505–3522.

[19] I. Hafeez, M. Antikainen, A. Y. Ding, and S. Tarkoma, “IoT-KEEPER: Detecting malicious IoT network activity using online traffic analysis at the edge,” *IEEE Trans. Netw. Service Manag.*, vol. 17, no. 1, pp. 45–59, Mar. 2020.

[20] M. Shafiq, Z. Tian, A. K. Bashir, X. Du, and M. Guizani, “CorrAUC: A malicious bot-IoT traffic detection method in IoT network using machine-learning techniques,” *IEEE Internet Things J.*, vol. 8, no. 5, pp. 3242–3254, Mar. 2021.

[21] P. Gope, Y. Gheraibia, S. Kabir, and B. Sikdar, “A secure IoT-based modern healthcare system with fault-tolerant decision making process,” *IEEE J. Biomed. Health Informat.*, vol. 25, no. 3, pp. 862–873, Mar. 2021.

[22] M. Adil, M. Khurram Khan, M. M. Jadoon, M. Attique, H. Song, and A. Farouk, “An AI-enabled hybrid lightweight authentication scheme for intelligent IoMT based cyber-physical systems,” *IEEE Trans. Netw. Sci. Eng.*, early access, Mar. 17, 2022, doi: 10.1109/TNSE.2022.3159526.

[23] I. A. Khan *et al.*, “XSRU-IoMT: Explainable simple recurrent units for threat detection in internet of medical things networks,” *Future Gener. Comput. Syst.*, vol. 127, pp. 181–193, Feb. 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167739X21003563

[24] A. Binbusayyis, H. Alaskar, T. Vaiyapuri, and M. Dinesh, “An investigation and comparison of machine learning approaches for intrusion detection in IoMT network,” *J. Supercomput.*, pp. 1–20, May 2022.

[25] R. M. S. Priya *et al.*, “An effective feature engineering for DNN using hybrid PCA-GWO for intrusion detection in IoMT architecture,” *Comput. Commun.*, vol. 160, pp. 139–149, Jul. 2020.

[26] R. Punithavathi, K. Venkatachalam, M. Masud, M. A. AlZain, and M. Abouhawwash, “Crypto hash based malware detection in IoMT framework,” *Intell. Autom. Soft Comput.*, vol. 34, no. 1, pp. 559–574, 2022.

[27] H. Xiong *et al.*, “On the design of blockchain-based ECDSA with fault-tolerant batch verification protocol for blockchain-enabled IoMT,” *IEEE J. Biomed. Health Informat.*, vol. 26, no. 5, pp. 1977–1986, May 2022.

[28] F. Farivar, M. S. Haghighi, A. Jolfaei, and M. Alazab, “Artificial intelligence for detection, estimation, and compensation of malicious attacks in nonlinear cyber-physical systems and industrial IoT,” *IEEE Trans. Ind. Informat.*, vol. 16, no. 4, pp. 2716–2725, Apr. 2020.

[29] Y. Sun, J. Liu, K. Yu, M. Alazab, and K. Lin, “PMRSS: Privacy-preserving medical record searching scheme for intelligent diagnosis in IoT healthcare,” *IEEE Trans. Ind. Informat.*, vol. 18, no. 3, pp. 1981–1990, Mar. 2022.

[30] R. Taheri, M. Shojafar, M. Alazab, and R. Tafazolli, “Fed-IIoT: A robust federated malware detection architecture in industrial IoT,” *IEEE Trans. Ind. Informat.*, vol. 17, no. 12, pp. 8442–8452, Dec. 2021.

[31] M. Amin, D. Shehwar, A. Ullah, T. Guarda, T. A. Tanveer, and S. Anwar, “A deep learning system for health care IoT and smartphone malware detection,” *Neural Comput. Appl.*, vol. 34, pp. 11283–11294, Nov. 2020.

[32] Y. Meidan *et al.*, “N-baiot—Network-based detection of IoT botnet attacks using deep autoencoders,” *IEEE Pervasive Comput.*, vol. 17, no. 3, pp. 12–22, Jul./Sep. 2018.

[33] S. M. Pudukotai Dinakarrao, H. Sayadi, H. M. Makrani, C. Nowzari, S. Rafatirad, and H. Homayoun, “Lightweight node-level malware detection and network-level malware confinement in IoT networks,” in *Proc. Design, Autom. Test Eur. Conf. Exhib. (DATE)*, Mar. 2019, pp. 776–781.

[34] J. Jeon, J. H. Park, and Y.-S. Jeong, “Dynamic analysis for IoT malware detection with convolution neural network model,” *IEEE Access*, vol. 8, pp. 96899–96911, 2020.

[35] A. Namavar Jahromi *et al.*, “An improved two-hidden-layer extreme learning machine for malware hunting,” *Comput. Secur.*, vol. 89, Feb. 2020, Art. no. 101655. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167404819301981

[36] J. Jeon, B. Jeong, S. Baek, and Y.-S. Jeong, “Hybrid malware detection based on bi-LSTM and SPP-net for smart IoT,” *IEEE Trans. Ind. Informat.*, vol. 18, no. 7, pp. 4830–4837, Jul. 2022.

[37] S. Khan and A. Akhunzada, “A hybrid DL-driven intelligent SDN-enabled malware detection framework for internet of medical things (IoMT),” *Comput. Commun.*, vol. 170, pp. 209–216, Mar. 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0140366421000347

[38] S. Liaqat, A. Akhunzada, F. S. Shaikh, A. Giannetsos, and M. A. Jan, “SDN orchestration to combat evolving cyber threats in internet of medical things (IoMT),” *Comput. Commun.*, vol. 160, pp. 697–705, Jul. 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0140366420312044

[39] M.-T. Luong, H. Pham, and C. D. Manning, “Effective approaches to attention-based neural machine translation,” 2015, *arXiv:1508.04025*.

[40] (2020). *Task 2: IoTMal2020-CDMC: IoT Malware Detection*. [Online]. Available: https://www.csmining.org/

[41] R. Ronen, M. Radu, C. Feuerstein, E. Yom-Tov, and M. Ahmadi, “Microsoft malware classification challenge,” 2018, *arXiv:1802.10135*.