Malware Traffic Identification in Internet of Things by Using a Convolutional Neural Network-Based Approach

Xin GE[1], Minnan YUE[2]*, Chunchang GAO[3]
Information Office, University of Shanghai for Science and Technology, Shanghai 200093, China[1, 3]
School of Energy and Power Engineering, University of Shanghai for Science and Technology, Shanghai 200093, China[2]

***Abstract*—The Internet of Things (IoTs) is the network from devices, sensors and physical tools, in order to exchange data by similar devices and the software through Internet. The expansion of IoTs in fields of smart healthcare, smart agriculture, smart home, smart city and similar fields has created a revolution on the human life. Due to the importance of IoTs, it is necessary to identify the malicious traffic for the maintenance of privacy, the stability of the network and the blocking of undesirable actions. In the current paper, a framework for the identification and the generation of novel samples from malware, based on the raw bytecode in the layer of the edge for IoTs is presented. Convolutional neural networks have been used to extract the features with high levels, and the boundary technique of the generative adversarial network has been applied to create novel samples from malware. Therefore, even with a few samples from malware, a considerable number of the prior unseen samples from malware can be identified with good accuracy. To get the short-term dependence and the long-term dependence from features, a method based on attention, which is a composition from CNN and LSTM, is used. The performance of the presented method is evaluated by a series of the experiments in dataset of IoT-23. The outcomes show that our approach obtains the foremost result with the greatest accuracy equal to 99.94%, compared to the most advanced machine learning methods.**

***Keywords*—Convolutional neural network (CNN); Internet of Things (IoT); malicious traffic; identification; long short term memory (LSTM)**

I. INTRODUCTION

IoTs is an interconnected system from the devices, the physical equipment, the computers, etc., which are distinguished from each other by a unique identifier, and benefit from the ability of the data transmission by using the Internet infrastructure. Currently, the technology and the scope of IoTs is expanding as day by day [1], and more devices are added to it every hour. According to the predictions until 2025, there will be more than 41 billion IoTs devices, while this number was about eight billion in 2019 [2, 3]. According to the report of the security company of Kaspersky, in first six months of 2019, further than 100 m attacks have been made on the IoTs devices, which is seven times more than in the first half of 2018 [4]. The attacks on the IoTs networks include four categories: the attacks of physical, the attacks of network, the attacks of software and the attacks of encryption [5]. These reported attacks are mainly the attacks of software and the attacks of network. The Mirai malware is an example from a widespread attack on the IoT devices that infected more than one million devices in 2016. Due to the security flaws in IoT devices, these devices are known as an attractive target for the cyber attackers. These devices are often manufactured with the aim of the reduction of the costs without serious security considerations, which makes them very vulnerable to the attacks; In many cases, even the default username and the default password of the IoT devices are not changed. In addition, the provided updates by the manufacturers of the IoTs devices are not applied to the mentioned devices, which in turn, leads to the creation of the more serious security flaws. In the latter years, many research has been done to identify and to deal with the done attacks on the IoT networks, but the attacks on the IoTs networks have also become the more sophisticated and the more efficient.

In the different phases of the identification, the contamination, the persistence and the operations [6], the attackers need to interact with the IoTs networks through the Internet and the network protocols; therefore, the investigation of the IoTs network traffic can help to the identification of the attacks and the compromised networks [7]. Although, the obfuscation of the network traffic by the attackers can cause the disturbances in the process of the identification of the attacks by the security experts, but, the deeper investigations into the structure and the nature of the different parts in the network traffic can result in an increment in accuracy and precision of the identification. Many researches have been conducted with the aim of the attacks identification on the IoTs infrastructure (especially the malware-based attacks) by using the IoTs network traffic analysis. In study [8], the authors use a hybrid method of the clustering of the fuzzy c-means and the scheme of the fuzzy concatenation, to evaluate the traffic of network and to identify the traffic of malicious. In study [9], the authors try to create an inference engine based on machine learning, by exploiting the signaling patterns and by helping the features such as the activity cycle in the network traffic. In [10], the authors have presented a model for identification of the scanning malware on the infrastructure of IoT, based on unique signatures (by focusing on the Mirai malware). In study [11], the researchers have presented a hybrid honey container approach, basis on existence of the authentication weaknesses in some services such as SSH and Telnet and based on the possibility of the command injection, to capture the malicious samples in the IoTs infrastructure.

Despite the efforts in the field of the identification of the attacks on the IoTs networks, this category of the security risks

is still expanding. One of the reasons for this is the inability to cover the existing identification methods and the number of the new unknown attacks. An efficient identification method should be able to adapt with the dynamic conditions for the identification of the zero-day unobserved attacks. In the current paper, a framework for the identification and the generation of the novel samples from malwares, basis on the raw bytecode in the layer of the edge for the IoTs networks, is presented. CNN has been used to extract the features with the high level, and the boundary technique of the generative adversarial network (GAN) has been applied to create the novel samples from the malware. Generative models have emerged as a crucial category of deep neural networks and are widely recognized. Introduced in 2014, GANs are a notable type of generative model with two primary uses in cybersecurity. Firstly, GANs can enhance the generalization and performance of machine learning models by creating new samples that mimic the original data. Secondly, they can generate adversarial examples to test and potentially compromise machine learning models. Therefore, even with a little samples from the malwares, a considerable number of the prior unseen samples from malwares can be identified by the good accuracy. To get the short-term dependence and the long-term dependence from the features, a method basis on the attention, which is a composition from CNN and LSTM, is used. The mechanism of the attention betters the performance of the method by increasing or by reducing the attention into the determined parts from features.

The key contributions of this research are: To enhance the stability of GAN training and improve the accuracy of malware detection classifiers, we introduce a new representation learning model using a CNN to extract high-level features from raw data; To bolster system security by enhancing model generalization and robustness, we increase the training set size by generating new malware sample signatures with a generative model known as Boundary seeking GAN. For precise malware detection, we propose an attention-based deep learning model that combines CNN and LSTM, which simultaneously considers both long-term and short-term dependencies in the input data and emphasizes critical features. The continuation of the paper is organized as the below. In Section II of article, the related researches to the analysis of the traffic and the identification of the malicious traffic in IoTs infrastructure, is presented. In Section III of article, our approach is described. In Section IV of article, results of our approach are reviewed. In Section V of article, the conclusions and the future works are given.

II. Related Works

With the extension of IoTs in recent years and the emergence of the security risks, several researches have been conducted in the field of the identification of these cyber threats, which each case has identified these attacks with the different levels of the accuracy and the speed. Some of these studies are mentioned in the below. In study [12], three models based on CNN for the detection of the malware in IoTs are investigated on 100 samples from malware of IoT in a 32-bit architecture, by comparing the various representations from data containing the trails, the assembly codes and the images. This comparison is done to detect the malicious files and the safe files. The findings show which either model performs as the fully good. In study [13], a 2-step combined scheme for the detection of the malware is proposed to support the devices against the ambiguous malware. This approach includes two stages for the identification of the Internet of Things malware. First, after doing the fixed analysis, opcodes are exploited and the files of the benign are identified with the use from the trained information through the two-way model of LSTM. Next, the running analysis is done in the files that are categorized as the benign in a nested virtual environment. After the extraction of the behavior information and the memory processing by the log of the behavior based on the variations in the system, the malware can be identified via trained approach.

In study [14], the IoTs malwares in the military domain have been investigated. Also, a model basis on the deep learning for the identification of the malware through the opcode sequence is presented. The proposed method transforms the opcodes to a space of the vector and applies a special model from space deep learning for the classification of the malicious programs. In addition, the strength of this method for the detection of the malware and its stability over the unwanted attacks for the insertion of the code have been investigated. In study [15], the authors have developed a method with the aim of the detection of the ransomware, which focuses on the traffic between the IoT devices and the outside world. The proposed method resides in SDN and analyzes the header of the CoAP packets and the TCP/IP packets. The proposed method is implemented in three steps. The first is collection of samples. The second is training of this approach, where the special features from the collected traffic in the previous step, are used. A combination from the naive bayes (NB) and PCA is proposed to identify ransomware, in second step and third step. The third is the detection and the moderation, where the ransomware attacks are detected by using the knowledge of the previous steps.

In study [16], the authors have proposed a software architecture as a network performance virtualizer, to deal with extension of malware in the networks of IoTs. To create a scalable intrusion detection system, an architecture of RNN is applied under the name of the LSTM learning. The input dataset consists of BoT-IoT, which after the pre-processing, is fed back to the input layers of the neural network. Basis on pre-processed data, model is trained and is used for the detection of the malicious traffic. In study [17], the authors have developed a software framework for the malware classification based on the MIPS architecture that uses the passive F-Sandbox and ML. This method uses network-based behavior for the collection of the information, in addition to the use of the intermediate application programming calls, the instruction tracing, the registry changes, the memory writes, etc. The machine learning algorithm includes the support vector machine (SVM) along with the scheme for the feature extraction of n-gram, and weight of the proposed trained model for the classification of five malware families is 97.44%.

In study [18], the authors have presented a non-invasive model, to derive the activities of the malicious from the malware at the IoTs network level through the analysis of the side-channel signals by using machine learning. First, the proposed method filters the signals of the devices, to achieve the signals of the suspicious. Next, by performing the fine-grained evaluation for the signals, it tries to derive the performed activities in devices. The proposed method specifies whether a

malware infection has occurred or not, by performing the correlation analysis on the detected activities. In study [7], the authors have presented a distributed method of the modular that can be applied for detection of the network-based activities on the IoTs infrastructure with a large scale. The proposed method uses ML for the classification of the traffic in the devices of the edge, the network database for the traffic vector from the feature and the security module from the policy. A scheme of ML is applied to classify the access gateway network traffic, with the use from the vector of the feature and the retrieved labels of the class from the database. Then, the method is transferred to a classifier of ML. If the novel malware is found, then, model is retrained and is compared to the current classifier for the checking of the performance.

III. The Proposed Method

In general, rather than the creation of a software in the first, the developers of the malware create the novel examples by making the lesser variations in the existed examples, to cause them harder for identification. Here, the novel signatures from the malware are generated by using GAN, based on a restricted number of the existed cases. Since the final goal of GAN is the estimation of the distribution of data, the created data is based on distribution of the existing malware. They are possibly analogous to the malware which can be published in future. After creating the novel samples, these data are added to set the training, which makes the network be trained by the varied samples. Fig. 1 displays the framework of our approach. First, the headers in the files are exploited. Next, the features of the abstract in the high level are extracted by CNN. Then, the novel signature from the samples of the malware is created by the boundary search GAN. This signature is added to the main set, to train the presented CNN. In the below sub-sections, the different steps of this proposed method are explained in the detail.

A. Header Extraction

The raw byte sequence of the header is used as a feature for the identification of the samples of the malware. The header is a little segment from file with the useful information, that it may be exploited under every conditions insomuch it is not encrypted [19]. In addition, it has been endeavoured to use minimum scope of the information. In this method, just three parts from the header of PE are used: OPTIONAL, MS-DOS and FILE. Basis on many tests in the bytes and the similar analyzes [20, 21, 22], it has been concluded that these fields of the header have the greatest influence in the malware identification on the files of PE. These parts are placed on the most files of PE and can be quickly exploited by the little overhead. Only header length and offset position are required for extraction of header. The required time for the extraction of headers is equal to 0.02 s, that is the much little, as compared with the extraction of the features like the opcode. Pursuant to the features of PE, the header of MS-DOS is equal to firstly 64 bytes from a file of PE. The information of FILE and OPTIONAL are placed on subsequent 248 bytes or subsequent 264 bytes, depending on whether software is for 32-bit devices or 64-bit devices [20]. Too, in the files of ELF, the header of ELF, that relies on 64-bit address or the 32-bit address and contains firstly 64 bytes or firstly 52 bytes from a file, is exploited [23]. Each extracted byte is transformed into a number in the decimal format, among 0 to 255, and in the next step, it is applied as the input in CNN. The header of ELF is a road map for representation of the structure of the file that contains the useful information like the type of the object file, the needed structure, the version, and so on. In this article, it has been tried to apply just the little parts from the executable file of the binary, and there is no requirement for the analyze or the exploitation. The needed time for the extraction of the header of ELF is equal to 0.0001 s, that is not significant.

B. High-Level Feature Extraction

One of the applications of the deep learning is capability to automatically learn the features of the conceptual. It catches a representation in the low level from the data and then, extracts a representation in the high level that is the further detachable for the classification. That's mean, the space of the conceptual is learned by the main space. Another benefit is the decrease in dimensions. If the dimensions in the space of the feature are larger, then further memory is needed for execution of the computations. Here, a deep model for the learning of the representation is implemented. Therefore, two models from the famous models, namely auto-encoder and CNN, which are used for the extraction of the feature, are implemented. Pursuant to the tests, CNN has performed better than the auto-encoder, for the feature extraction. Therefore, CNN has been used in the proposed method. In the presented DNN, the layer of the embedding is applied to transform the distinct vectors of the feature into the continuing vectors [24].

There are methods for the execution of this task, including the encoding of one-hot. In the encoding of one-hot, the creation and the update of a vector is rapid and simple. Nevertheless, sometimes, its dimensions can enhance as dramatic. Also, it cannot train the relations among the vectors of the feature. However, the method of the embedding can train the logical relations among the vectors of the feature. Next, 2 1D convolution layers are used, that can exploit the dependence among features. Conv-1D, according to its framework, has much the less complexity of the computational against Conv-2D. Therefore, it can be trained with the great speeds and without ever strong hardware. Thus, it can be applied on the applications in the real time with the low cost. Another benefit of Conv-1D is that the complex sequential data structures can be learned even with a shallow architecture [25]. After every layer of the convolution, a layer of the max-pooling is adopted, that is a common solution for the down-sampling. The layer of the pooling is responsible for the collection of the useful features of the abstract, to avoid the overfitting or to decrease the complexity of the computational [24]. Eventually, a layer of the dense by 32 neurons is adopted, that is the newly extracted feature vector.

C. Generation of Data and Oversampling of Training Set

In the proposed method, the boundary search GAN is used to create the signatures of novel samples from malwares. The generator produces the samples basis on the random distribution like Gaussian. It endeavors to transmit the random distribution into distribution of the main samples from malwares. Next, the generated samples and the actual samples are provided on discriminator, to be categorized in the respective categories. After the training of network, the generator learns the original

distribution of the malware, and can generate new signatures from the samples of the malware. Eventually, the created data are aggravated in the main dataset. The proposed approach is the stronger against the noise and the data manipulation, because of this over-sampling. The framework of generator and the discriminator is displayed on Fig. 2. Because the features in the high level that CNN extracts as the samples of the training in previous step, are used, the process of the training is the very rapider and the further permanent than when the main features are applied. Too, a single-way technique for the smoothing of the label is applied, and the positive label of the class is shifted by 0.9. Pursuant to Goodfellow [26], this method betters the efficiency of discriminator over the tricks of the generator.

D. Malware Identification

Eventually, a DNN is proposed for the malware identification, which the general framework is shown on Fig. 3. In it, there are two 1-D layers of the convolutional, which extract the multiple features of the local from the long trails of the input. The next maps are computed basis on the below equation, where input is represented by f by length n, and kernel is represented by h with length m:

$$(f * g)(i) = \sum_{j=1}^{m} g(i) \cdot f(i - j + \frac{m}{2}) \quad (1)$$

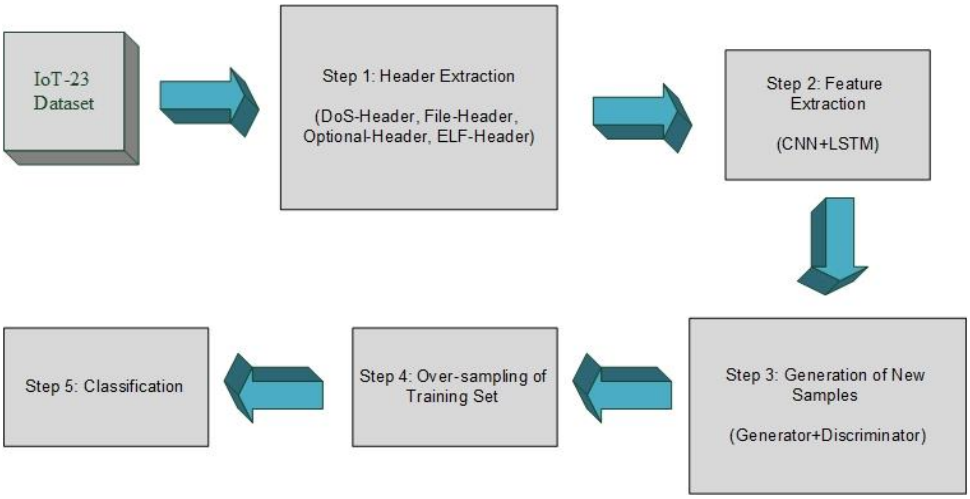

Fig. 1. The general structure of our approach.

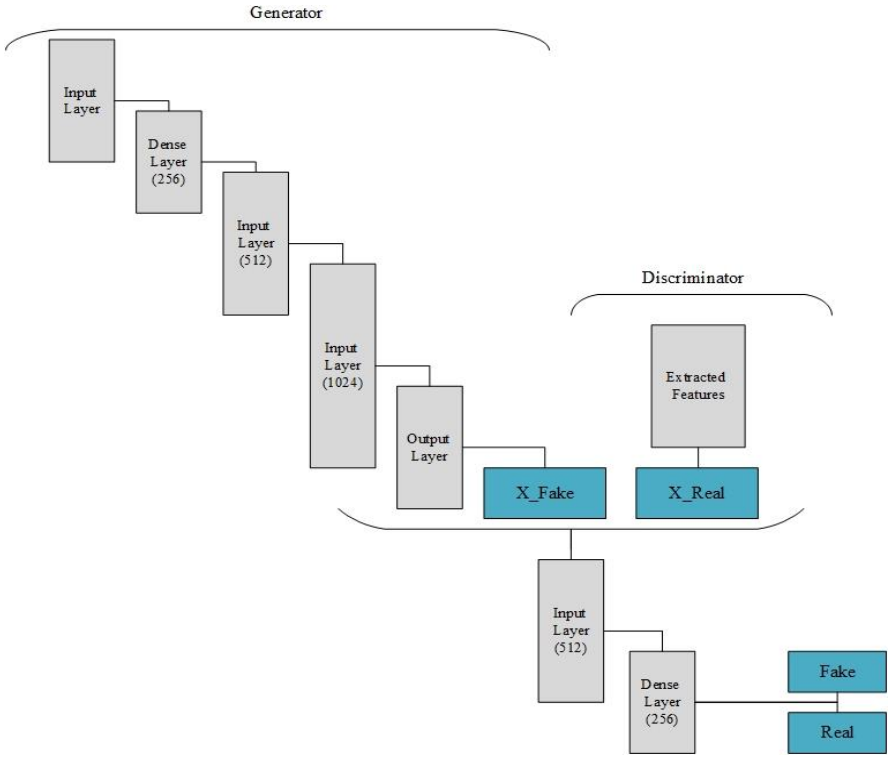

Fig. 2. The overall framework of generator and discriminator.

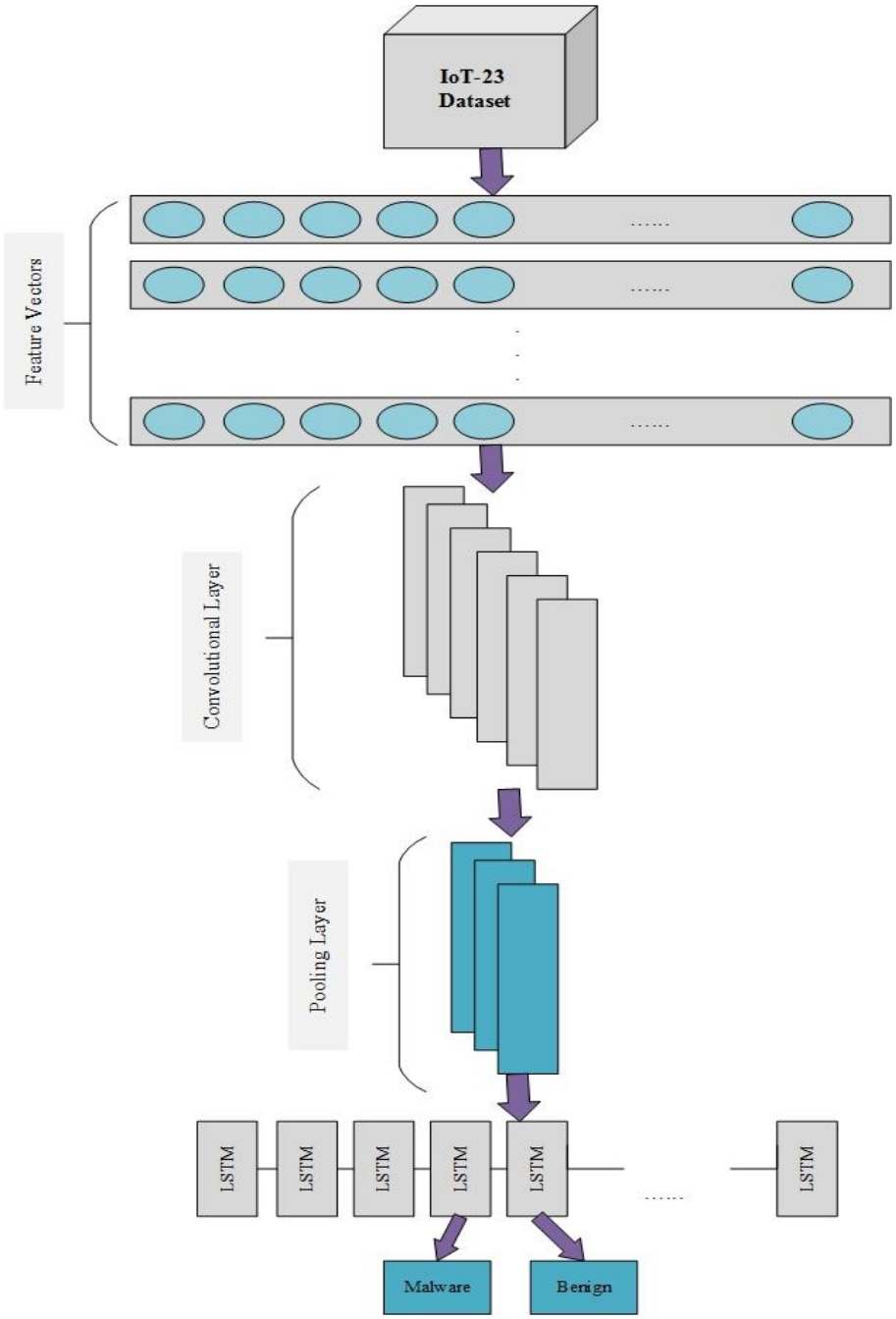

Fig. 3. Our designed network structure for the malware identification.

After every layer of the convolution, there are the pooling layer and the dropout layer, to avoid the overfitting [24]. The dropout is a method which randomly drops a number of the neurons from every layer, with a fixed rate in the training of the network. Therefore, the approach does not rely on a special set from the neurons and the weights. Next, a layer of LSTM is adopted, that decreases the influence of the vanishing slope and smooth the road [27]. LSTM includes the internal manners which can regulate the flow of data.

$$f_t = \sigma(W_f \cdot [h_{t-1}.x_t] + b_f) \quad (2)$$

$$i_t = \sigma(W_i \cdot [h_{t-1}.x_t] + b_i) \quad (3)$$

$$o_t = \sigma(W_o \cdot [h_{t-1}.x_t] + b_o) \quad (4)$$

$$C_t = f_t * C_{t-1} + i_t * \tanh(W_c \cdot [h_{t-1}.x_t] + b_C) \quad (5)$$

$$h_t = o_t * \tanh(C_t) \quad (6)$$

f_t represents the gate of the forget, which is responsible for the removal of the information that the state cell does not need it. i_t represents the gate of the input, which determines what novel data can be added to cell. o_t represents the gate of the output, which specifies output of current cell. C_t indicates the state of the cell, and h_t indicates hidden state, that is output from the layer of LSTM. To decrease the time of the execution, the layers of the convolutional and the layers of the max-pooling

have been applied to exploit the features from the long trails and to send these features to the layer of LSTM. The composition of LSTM and CNN displays the short-term dependencies and the long-term dependencies as simultaneously [28]. Then, the memory takes into account correlations between the features. Next, the mechanism of the attention [29] is applied to specify the relationship among the various locations of a trail. The main stimulant is the increase of the accuracy, by enhancing or by reducing focus and attention on the specified parts from the trails of the input. The mechanism of the attention emulates the manner of the vision in the human. When the mechanism of the vision detects a case, it does not seek the whole scene and focuses on a specified part. The attention learns relationship among the features and their effect in the identification of the goal. The stages are described in below relations. h_i displays the hidden states in the layer of LSTM. w_0 and w_1 displays the matrix of the weights. A NN is trained, to allocate the grades to the connected inputs. When the grades are computed, the weights of the attention are produced, by doing function of Softmax. The sum of weights must be 1. Then, this vector is multiplied by the vector of the input, to use weights on output. Then, input is hidden state in the layer of LSTM. Therefore, a weight is computed for every state, which displays its significance in a classification work. Also, it causes that the approach be the stronger over the noise. out is fed to a layer of the fully-connected by the activation of Softmax, to perform the classification.

$$out = \sum_{i=1}^{T} a_i h_i \tag{7}$$

$$\bar{h} = \frac{1}{T}\sum_{i=1}^{T} a_i h_i \tag{8}$$

$$a_i = \tan(w_0^T h_i + w_1^T \bar{h}) \tag{9}$$

Finally, it should be mentioned that the used optimizer is Adam, and the size of the batch is adjusted to 80. The function of the loss is the binary entropy-cross. CNN is trained as a module from the extraction of the feature in 15 iterations, and the proposed network is trained in 1200 iterations. The details of the discriminator and the generator are displayed in Fig. 4 and Fig. 5. Too, the proposed DDN for the malware identification is trained in 20 iterations.

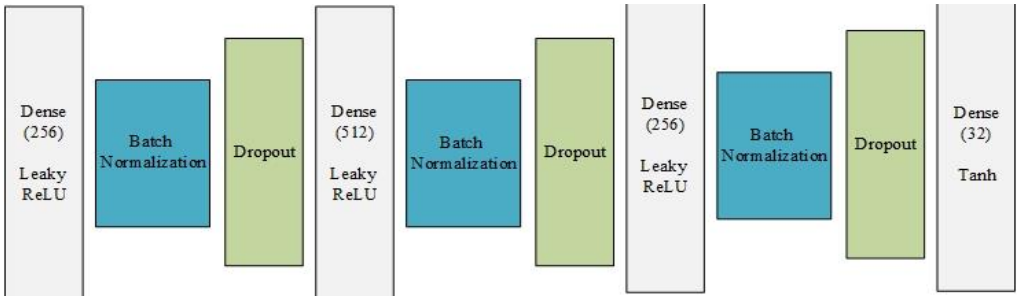

Fig. 4. The framework of generator.

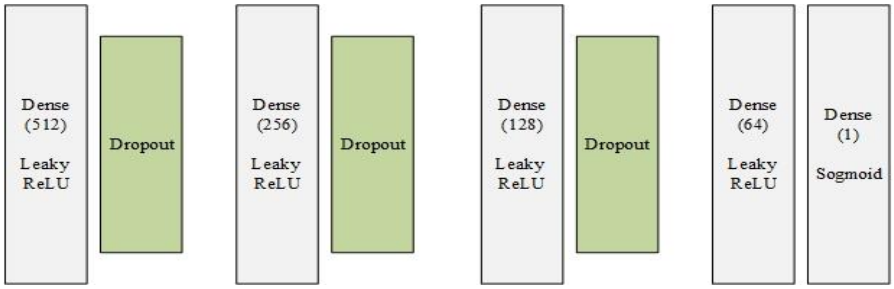

Fig. 5. The framework of discriminator.

IV. Experiments and Evaluation of Results

Here, details of dataset and experiments and results are provided. Python has been applied for implementation of experiments. Our approach is implemented in a computer by 8G RAM and Core(TM) i7 CPU 3.0 GHz Intel(R). CNN is implemented on GPU, and the card of the graphics is NVIDIA GEFORCE 840M.

A. Dataset

On the current article, dataset of IoT-23 [30], which is an IoTs network traffic dataset, is used. This dataset consists of three benign recordings and 20 malware recordings, which are gathered in 2018 to 2019 at Czech Republic, Stratosphere Lab, CTU University. For the scenarios of the benign, the traffic of the network is gathered by three real IoTs devices: a smart LED light bulb, a device of Amazon Echo and a smart lock of the door. It enables others to perceive the usual behavior of IoTs networks. According to the big size of IoT-23, seven scenarios of the malware and three scenarios of the benign, which comprise more than 1200000 Zeek flows, are selected. The dataset has 21 features, which one of them is the class label. The dataset is randomly divided into three segments: 70% for the training, 10% for the validation and 20% for the test.

B. Evaluation Criteria

The below criteria are applied, to analyze efficiency of different methods: 1) TP, that displays number of the flows of the benign which are accurately classified; 2) TN, that displays the samples of the malware which are accurately identified; 3)

FP, that displays the flows of the benign which are incorrectly identified as the malware; 4) FN, that displays the samples of the malware which are incorrectly identified as the benign; 5) Accuracy displays the percent of the precise identifications: $Accuracy = \frac{TN+TP}{TN+TP+FN+FP}$; 6) Precision displays a proportion from the true positives to the total precise identifications: $Precision = \frac{TP}{FP+TP}$; 7) Recall displays the expected amount of the malwares which are accurately identified: $Recall = \frac{TP}{FN+TP}$; 8) F1-Score displays a weighted average for Precision and Recall, which its foremost value is equal to 1 and its worst value is equal to 0: $F1 - Score = 2 \times \frac{Recall \times Precision}{Recall+Precision}$.

C. Results

Here, the findings of our approach are presented on IoT-23. In addition, a comparison among the efficiency of our approach, the advanced machine learning models, the models basis on deep learning and the ensemble learning malware detection models is presented. At first, the efficiency of our approach is analyzed as separately and by using different criteria, on the selected dataset. As shown in Table I, our approach the good efficiency, according to all performance criteria. Also, the loss values of the proposed approach have been drawn, to analyze the influence of the exploited features by the training of CNN and to check how of convergence of the generator and the discriminator. As displayed on Fig. 6, when the main bytecodes are used for training of the proposed approach, the error among the generator and the discriminator is not moderated and is fastly changed. Nevertheless, as shown on Fig. 7, the rate of the error is moderated with the use from the exploited feature, and the values of the error is not rapidly varied.

To make a comprehensive analysis, the efficiency of our approach is compared to the advanced machine learning models, the malware detection models basis on deep learning and the ensemble learning models. The comparison results are shown in Table II. For the machine learning methods, an advanced machine learning algorithm is selected and is implemented from each article, which is shown in the column "Model" in Table II. The results show that our approach, has a higher efficiency than the similar models. The highest Accuracy is equal to 99.94%, which is related to the proposed method. In addition, according to Precision and F1-Score, the rate of the improvement is significant.

Fig. 8 to Fig. 11 graphically show the comparison among the performance of different models and our approach based on the different criteria. As can be viewed, in all experiments, our approach performs better than similar machine learning-based models, the models basis on deep learning, and the ensemble learning models. The findings confirm which our presented approach outperforms than similar works, according to all different criteria. Briefly, our proposed approach obtains to a passable efficiency in the classification. Also, this improvement in the efficiency is due to the use from the mechanism of the attention, which increases influence of the related features on classification and increases accuracy of our approach. Since our approach has reliable efficiency (even in the restricted data for the training), it can be applied on the configurations with the inadequate unbalanced data. In addition, the proposed approach handles the consecutive data, which can be used to identify the malware basis on the traffic of the network, the opcode, etc.

D. Discussion

The results from this study clearly show that model performance has significantly improved after applying GAN for oversampling. We increased the positive class (malware) using samples generated by GAN and the negative class (benign) using original Windows software. The performance enhancement is evident. Deep networks are advantageous as they can extract conceptual spaces from data. Our proposed deep network, which combines CNN and LSTM, considers both short-term and long-term dependencies simultaneously. Additionally, selecting the optimal structure and fine-tuning network hyper-parameters after extensive experimentation were crucial. Even with a limited number of training samples, the proposed model can detect malware with reasonable accuracy. In contrast, similar methods require more training samples due to their network structure and image-based input type. They also perform poorly in the feature engineering phase with limited data and only consider short-term dependencies between features.

TABLE I. THE EXPERIMENTAL RESULTS OF THE PROPOSED APPROACH ON IOT-23

Model	Accuracy	Precision	Recall	F1-Score
Our Model	99.94	99.89	99.97	99.95

TABLE II. THE COMPARISON AMONG THE EFFICIENCY OF LATTER MODELS FOR THE DETECTION OF THE MALWARE AND THE PROPOSED APPROACH ON IOT-23

Type	Model	Accuracy	Precision	Recall	F1-Score
Machine Learning Methods	Method in [31]	97.96	97.12	99.13	98.76
	Method in [32]	98.99	98.84	99.14	99.05
	Method in [33]	96.74	93.68	92.89	93.27
Deep Learning Methods	Method in [34]	99.12	98.97	99.25	99.15
	Method in [35]	98.84	98.67	99.11	98.88
	Method in [36]	96.44	97.14	96.87	95.97
Ensemble Learning Methods	Method in [37]	98.59	96.81	91.92	95.77
	Method in [38]	94.98	95.11	96.12	95.34
	Method in [39]	99.17	99.39	98.77	99.12
Our Method		99.94	99.89	99.98	99.95

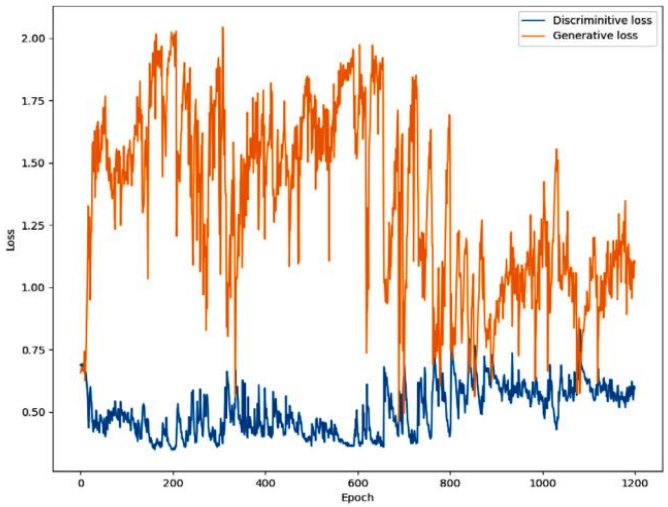

Fig. 6. The loss of generator and discriminator, when the main header is applied for training of the proposed approach.

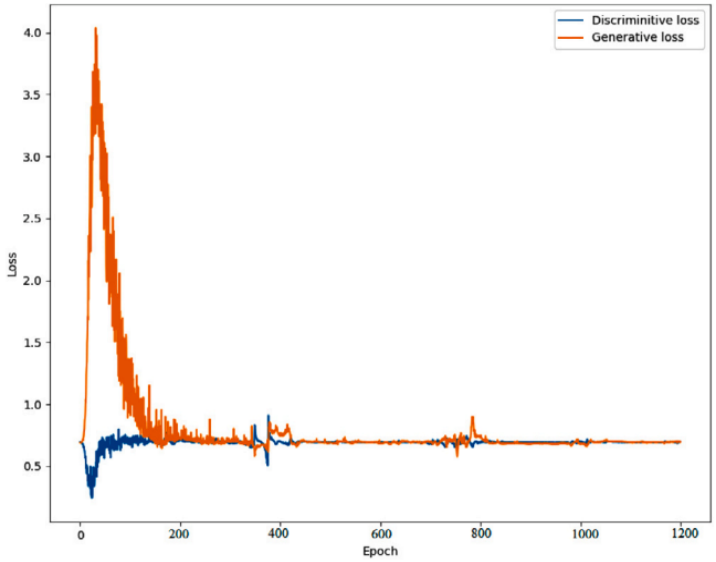

Fig. 7. The loss of generator and discriminator, when the features of CNN are applied for the training of the proposed approach.

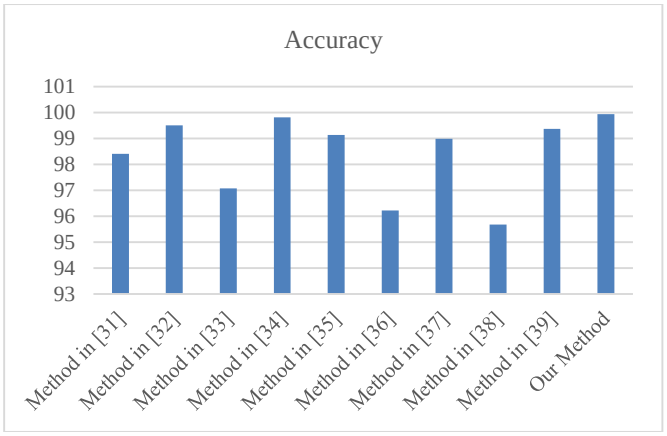

Fig. 8. The results of accuracy on IoT-23.

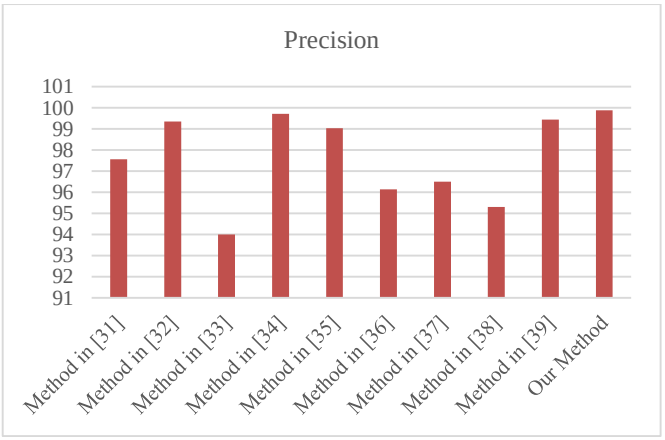

Fig. 9. The results of precision on IoT-23.

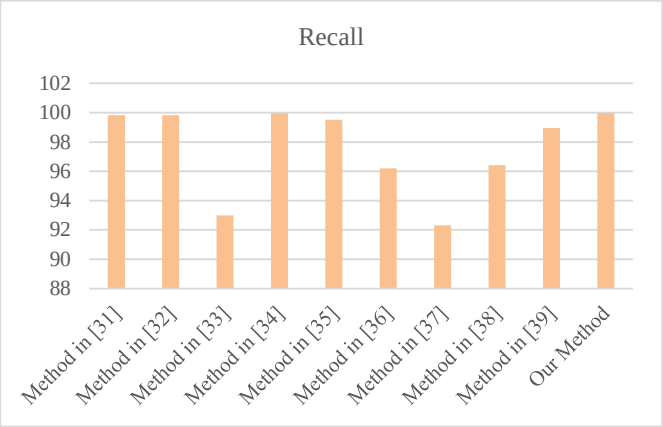

Fig. 10. The results of recall on IoT-23.

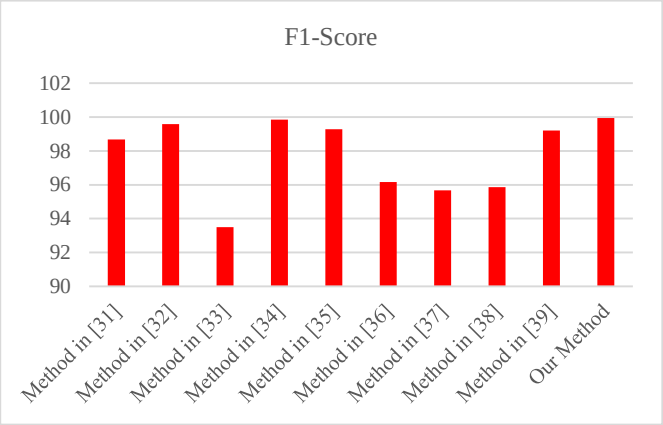

Fig. 11. The results of F1-score on IoT-23.

V. Conclusions and Suggestions

The ever-increasing expansion of the IoTs devices and the lack of the proper security policies at the time of the production on them, have turned this category of the equipment into the potential targets for the cyber attackers and the computer malware developers. The resource limitation in the IoTs devices is the main challenge in the implementation of the security mechanisms. On the current article, a novel approach for the identification and the generation of the novel samples from malware in IoTs, basis on the main header bytes, is provided. Too, a model for the automatic learning of the representations is implemented, to extract the space of the conceptual from the main data that betters the process of the training in the boundary search GAN and increases the accuracy. The findings display which accuracy of classifiers is significantly bettered, that denotes the effect of the created data in generalization and the strength of classifiers. Too, a DNN is used that simultaneously catches the dependencies of the local and the global in the data, to identify the samples of the malware. The findings display which our approach outperforms than similar approaches, according to all criteria. In future works, it is possible to evaluate the malicious traffic in the IoTs networks by using other features of the network packets and by developing other deep learning models, along with the activation algorithms and the different cost functions. Also, GANs can be used to generate and to identify the Hostile samples. The Hostile samples are between most important attacks which menace the models of ML. In addition, in the future, the malware samples can be produced and can be analyzed with the use from the similar manners, such as the running on an implicit environment.

Acknowledgment

This work was supported by China University Industry-University-Research Innovation Fund-New Generation Information Technology Innovation Project (Grand No. 2022IT144).

References

[1] B. Kaur and V. Dhir, “Internet of things: Vision, challenges and future scope,” Int. J. Adv. Res. Comput. Sci., vol. 8, no. 4, pp. 40–43, 2017.

[2] T. Fougeroux, A. Douyere, P. O. L. de Peslouan, N. Murad, S. Oree, and J.-L. Dubard, “Circuit Model of Rectennas Array for Estimating Microwave Energy Harvesting in Presence of Mutual Coupling Between Elements,” in 10ième Journées Nationales sur la Récupération et le Stockage de l’Energie (JNRSE 2021), 2021, p. 2.

[3] “Internet of Things Report.” https://www.businessinsider.com/internet-of-thingsreport (accessed Nov. 13, 2021).

[4] “Things just got real: 61% of businesses already use IoT platforms despite security risks | Kaspersky.” https://www.kaspersky.com/about/pressreleases/2020_things-just-got-real-61-of-businessesalready-use-iot-platforms-despite-security-risks (accessed Nov. 13, 2021).

[5] J. Deogirikar and A. Vidhate, “Security attacks in IoT: A survey,” in 2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), 2017, pp. 32–37.

[6] C. McCormack, “Five stages of a web malware attack.” Abingdon. Retrieved from https://www. sophos. com/en-us/medialibrary/Gated …, 2016.

[7] A. Kumar and T. J. Lim, “EDIMA: early detection of IoT malware network activity using machine learning techniques,” in 2019 IEEE 5th World Forum on Internet of Things (WF-IoT), 2019, pp. 289–294.

[8] 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. Serv. Manag., vol. 17, no. 1, pp. 45–59, 2020.

[9] A. Sivanathan, “Iot behavioral monitoring via network traffic analysis,” arXiv Prepr. arXiv2001.10632, 2020.

[10] A. Kumar and T. J. Lim, “Early detection of Mirailike IoT bots in large-scale networks through subsampled packet traffic analysis,” in Future of Information and Communication Conference, 2019, pp. 847–867.

[11] B. Wang, Y. Dou, Y. Sang, Y. Zhang, and J. Huang, “IoTCMal: Towards a hybrid IoT honeypot for capturing and analyzing malware,” in ICC 2020- 2020 IEEE International Conference on Communications (ICC), 2020, pp. 1–7.

[12] K. D. T. Nguyen, T. M. Tuan, S. H. Le, A. P. Viet, M. Ogawa, and N. Le Minh, “Comparison of three deep learning-based approaches for IoT malware detection,” in 2018 10th international conference on Knowledge and Systems Engineering (KSE), 2018, pp. 382–388.

[13] S. Baek, J. Jeon, B. Jeong, and Y.-S. Jeong, “Twostage hybrid malware detection using deep learning,” Human-centric Comput. Inf. Sci., vol. 11, no. 27, pp. 10–22967, 2021.

[14] A. Azmoodeh, A. Dehghantanha, and K.-K. R. Choo, “Robust malware detection for internet of (battlefield) things devices using deep eigenspace learning,” IEEE Trans. Sustain. Comput., vol. 4, no. 1, pp. 88–95, 2018.

[15] A. Wani and S. Revathi, “Ransomware protection in IoT using software defined networking,” Int. J. Electr. Comput. Eng., vol. 10, no. 3, pp. 3166–3175, 2020.

[16] N. Guizani and A. Ghafoor, “A network function virtualization system for detecting malware in large IoT based networks,” IEEE J. Sel. Areas Commun., vol. 38, no. 6, pp. 1218–1228, 2020.

[17] T. N. Phu, K. H. Dang, D. N. Quoc, N. T. Dai, and N. N. Binh, “A novel framework to classify malware in mips architecture-based iot devices,” Secur. Commun. Networks, vol. 2019, 2019.

[18] F. Ding et al., "DeepPower: Non-intrusive and deep learning-based detection of IoT malware using power side channels," in Proceedings of the 15th ACM Asia Conference on Computer and Communications Security, 2020, pp. 33–46.

[19] M. Farrokhmanesh, A. Hamzeh, Music classification as a new approach for malware detection, J. Comput. Virol. Hacking Tech. 15 (2) (2019) 77–96.

[20] E. Raff, J. Sylvester, C. Nicholas, Learning the pe header, malware detection with minimal domain knowledge, 2017, pp. 121–132.

[21] T. Rezaei, A. Hamze, An efficient approach for malware detection using pe header specifications, in: 2020 6th International Conference on Web Research (ICWR), IEEE, 2020, pp. 234–239.

[22] E. Raff, J. Barker, J. Sylvester, R. Brandon, B. Catanzaro, C. Nicholas, Malware detection by eating a whole exe, 2017, arXiv preprint arXiv:1710.09435.

[23] T. Committee, et al., Tool interface standard (tis) executable and linking format (elf) specification version 1.2, 1995.

[24] L. Liu, B. Wang, Automatic malware detection using deep learning based on static analysis, in: International Conference of Pioneering Computer Scientists, Engineers and Educators, Springer, 2017, pp. 500–507.

[25] S. Kiranyaz, O. Avci, O. Abdeljaber, T. Ince, M. Gabbouj, D.J. Inman, 1d convolutional neural networks and applications: A survey, 2019, arXiv preprint arXiv:1905.03554.

[26] I. Goodfellow, Nips 2016 tutorial: Generative adversarial networks, 2016, arXiv preprint arXiv:1701.00160.

[27] F.A. Gers, J. Schmidhuber, F. Cummins, Learning to forget: Continual prediction with lstm, 1999.

[28] T. Liu, J. Bao, J. Wang, Y. Zhang, A hybrid cnn–lstm algorithm for online defect recognition of co2 welding, Sensors 18 (12) (2018) 4369.

[29] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, in: Advances in Neural Information Processing Systems, 2017, pp. 5998–6008.

[30] Parmisano A, Garcia S, Erquiaga M (2020) A labeled dataset with malicious and benign iot network trafc. Stratosphere Laboratory, Praha, Czech Republic.

[31] Dzulqarnain D (2019) Investigating IoT malware characteristics to improve network security. University of Twente, Netherlands.

[32] Banerjee M, Samantaray S (2019) Network trafc analysis based IoT botnet detection using honeynet data applying classifcation techniques. Int J Comp Sci Inf Secur (IJCSIS) 17(8).

[33] Jamal A, Hayat MF, Nasir M (2022) Malware detection and classifcation in IoT network using ANN. Mehran Univ Res J Eng Technol 41(1):80–91.

[34] Ahmed AA, Jabbar WA, Sadiq AS, Patel H (2020) Deep learning-based classifcation model for botnet attack detection. J Ambient Intell Human Comput 1–10.

[35] Parameswaran Lakshmi S (2020) A lightweight 1-D CNN model to detect android malware on the mobile phone. National College of Ireland, Dublin.

[36] Xing X, Jin X, Elahi H, Jiang H, Wang G (2022) A malware detection approach using autoencoder in deep learning. IEEE Access 10:25696–25706.

[37] Saharkhizan M, Azmoodeh A, Dehghantanha A, Choo K-KR, Parizi RM (2020) An ensemble of deep recurrent neural networks for detecting IoT cyber attacks using network trafc. IEEE Internet Things J 7(9):8852–8859.

[38] Sahu AK, Sharma S, Tanveer M, Raja R (2021) Internet of things attack detection using hybrid deep learning model. Comput Commun 176:146–154.

[39] Nguyen GL, Dumba B, Ngo Q-D, Le H-V, Nguyen TN (2022) A collaborative approach to early detection of IoT Botnet. Comput Electr Eng 97:107525.