A Novel Web Attack Detection System for Internet of Things via Ensemble Classification

Chaochao Luo, Zhiyuan Tan, Geyong Min, Jie Gan, Wei Shi, and Zhihong Tian*

***Abstract*—Internet of things (IoT) has become one of the fastest-growing technologies and has been broadly applied in various fields. IoT networks contain millions of devices with the capability of interacting with each other and providing functionalities that were never available to us before. These IoT networks are designed to provide friendly and intelligent operations through big data analysis of information generated or collected from an abundance of devices in real time. However, the diversity of IoT devices makes the IoT networks environments more complex and more vulnerable to various web attacks compared to traditional computer networks. In this paper, we propose a novel Ensemble Deep Learning based Web Attack Detection System (EDL-WADS) to alleviate the serious issues that IoT networks faces. Specifically, we have designed three deep learning models to first detect web attacks separately. We then use an ensemble classifier to make the final decision according to the results obtained from the three deep learning models. In order to evaluate the proposed WADS, we have performed experiments on a public dataset as well as a real-word dataset running in a distributed environment. Experimental results show that the proposed system can detect web attacks accurately with low false positive and negative rates.**

***Index Terms*— IoT, Deep Learning, Ensemble Classifier, Web Attack Detection**

I. Introduction

As one of the fastest-growing and widely used technologies on internet, Internet of things (IoT) extends the edge of the Internet by connecting additional terminal devices and facilities on the edge of the network. Specifically, IoT contains millions of devices with the capability of interacting with each other and providing great convenience for us. Via IoT technology, smart cities, smart home, smart medical treatment, smart agriculture and other smart fields are emerging. Our ways of life and work are becoming easier, more efficient, more interesting and more convenient.

There are millions of IoT devices all over the world, some of which are visible to us while others are not. The data collected from these devices and stored in datacenters contain vast amounts of information, which may contain individuals' private information. More visible and invisible threats are emerging and causing irrecoverable damages. Due to the high concentration of various information, attackers often select storage and service servers as a primary attack target. Once the attackers gain access to the central severs, data breaches are inevitable. Furthermore, the local storage and computing limitations of IoT devices prevent them from detecting and defending against potential web attacks. A minor security threat has the potential to cause severe damage to IoT networks. Therefore, there is no doubt that ensuring the security of IoT networks is of great significance to the success of IoT applications. Compared with traditional computer networks, there are more terminal devices and traffic in IoT networks, which make IoT network security issues more complex and troublesome [4]. Recent works covering web attack detection systems have shown a great capacity for the protection of traditional networks. However, these systems have faced severe challenges when utilized in IoT networks. Thus, there is an urgent need for research into more progressive systems to protect IoT networks from various web attacks [7-8, 17].

As web attacks grow rapidly in sophistication and diversity, researchers of network security are actively exploring new security technologies based on deep learning [1-3]. While traditional web attack detection technologies show weaknesses in big data environment, the rise of deep learning provides novel solutions to security problems in such environments. Deep learning applications, based on big data analysis, show superior capacity for detecting aggression through massive traffic flow. These deep learning solutions have helped to advance and facilitate the development of IoT network security.

In this study, we propose a novel Web Attack Detection System (WADS) for IoT networks, based on ensemble deep learning. Specifically, the proposed EDL-WADS takes advantage of deep learning models to analyze Uniform Resource Locator (URL) requests in the network traffic and identify anomalous requests within which web attack payloads are attached. In our approach, three deep learning models are employed to each learn relative features hidden in the queries. We use different methods to process and transform URL requests into different types of representations in order to exploit the advantages from a variety of deep learning models.

This article was supported in part by the National Natural Science Foundation of China under Grant U20B2046, in part by the National Key research and Development Plan under Grant 2018YFB0803504, in part by the Guangdong Province Key Research and Development Plan under Grant 2019B010137004. Paper no. TII-20-3138. (Corresponding author: Zhihong Tian.)

C. Luo, and J. Gan are with the ADLab of Venustech, Beijing 100001, China (e-mail: luochaochaoforaffair@gmail.com; charles2gan@gmail.com).

Z. Tan is with the School of Computing, Edinburgh Napier University, Edinburgh EH10 5DT, U.K. (e-mail: Z.Tan@napier.ac.uk).

G. Min is with the Department of Computer Science, University of Exeter, Exeter EX4 4QF, U.K. (e-mail: G.Min@exeter.ac.uk).

W. Shi is with the School of Information Technology, Carleton University, Ottawa K1S 5B6, Canada (e-mail: WeiShi@cunet.carleton.ca).

Z. Tian is with the Cyberspace Institute of Advanced Technology, Guangzhou University, Guangzhou 510006, China (e-mail: tianzhihong@gzhu.edu.cn).

Moreover, we employ an ensemble classifier to conduct a comprehensive analysis of the results of these three deep learning models. The ensemble classifier is designed to allow EDL-WADS to overcome the weaknesses of the individual classifiers and combine their advantages to improve the detection performance.

The contributions of this paper are summarized as follows:

- We propose EDL-WADS, a novel ensemble deep-learning-based system that can detect anomalous queries in which malicious codes are attached in an IoT network.
- We utilize a group of deep learning models to produce different representations of URL requests in order to exploit the advantages from a variety of classification.
- An ensemble classifier is utilized in EDL-WADS to improve the detection performance by combining results from different classifiers based on Multi-Layer Perceptrons (MLP).
- We compare our proposed approach with several existing approaches deployed in a distributed environment. Our experimental results confirm the effectiveness and superiority of EDL-WADS in detecting IoT web attacks in real time.

The remainder of this paper is organized as follows: In Section 2, we present the state-of-the-art works in this research field. We introduce the proposed system in Section 3. In Section 4, we present the experimental results and their analysis. Finally, we draw conclusions in Section 5.

II. Related work

After the breakthrough of artificial intelligence technology, deep learning has been widely used by researchers in the field of network security. There is a great deal of researches have been done focused on web attack detection based on deep learning [5, 9-12]. And it seems that security detection technology based on artificial intelligence is gradually becoming the primary direction techniques. As a matter of fact, methods for web attack detection based on deep learning are driven by big data analysis. In this way, deep learning models can analyze inputs by extracting these useful features and learn a pattern from these features by iterative training. Depend on deep learning techniques, the web attack detection techniques make a progressive improvement in detection performance [16, 21-23]. At present, the contributions of existing related works are mainly reflected in two aspects: one is the method applied to analyze URL requests and transform them into vectors and the other is the deep learning model utilized to learn features and detect web attacks. We summarize these three types of methods for URL analysis below:

- Statistical characteristics based on matching and counting anomalous words or punctuations from raw traffic are most widely used to represent URL requests, such as the length of URL requests, the anomalous words of punctuations in the requests, the types of anomalous words and the number of parameters.
- Representing URL requests based on traditional semantic analysis and syntactic analysis from raw data has become a popular way in the field of web attack detection. Features extracted from semantic analysis and syntactic analysis contains the depth of the syntax tree, the number of roots in the syntax tree, the number of leaf nodes in the syntax tree, etc.
- The method of analyzing URL requests and transforming them into vectors automatically shows its superior capability of representing URL requests accurately. And it has become the state-of-the-art method in the field of web attack detection.

The method based on deep learning makes full use of the advantages of big data analysis and can detect web attacks more comprehensively and accurately. Ma *et al*. [18] used static features and evaluated the methods with Naive Bayes model, support vector machine (SVM) and Logistic Regression (LR). The results show the deep learning model's capacity of identifying web attack through these static features. Also, Kar *et al*. [2] proposed a system for web attack detection, in which the method based on statistical characteristics is used to represent URL requests and a novel deep learning model is used to do classification task. The results achieved a high accuracy of 96.37%. Compared with the traditional detection method, deep learning approaches based on statistical characteristics makes a significant increase in the result accuracy. However, there are two drawbacks of this method: first, it costs a lot in defining the special dictionary; second, the dictionary cannot include all anomalous words of expressions. Consequently, the hackers can bypass the matching rules with constantly changing payloads.

Actually, features extracted by the method based on semantic analysis uses their statistical characteristics. These features depend on statistical characteristics of syntax trees generated by semantic analysis and syntactic analysis instead of raw requests. Lee *et al*. [19] proposed a novel method to detect SQL injection with removing values of SQL queries and comparing them with predetermined syntactic rules. Compared with other approaches, the results show that the proposed method is simpler and more effective. The study in [11] used semantic tools to get a syntax tree from URL requests and defined various of statistical characteristics based on the syntax tree. Experimental results showed that their approach achieved promising performs in web attack detection. Compared with the former method, the second method reduces manual intervention to some extent and overcomes the disadvantage of the first method. However, the second method doesn't show significant improvement in the performance of web attack detection.

As for the third method, it has been state of the art method in the field of web attack detection. Compared with the first two methods, this method can analyze URL requests and transform them into vectors automatically, and overcome the disadvantages of the first two methods with significant

improvement in the performance. Kar *et al*. [13] proposed a method based on digraph to analyze and transform URL requests automatically. The results show that the proposed method performed well and obtained the highest accuracy at 99.63%. Also, Yong *et al*. [20] proposed a new automatic method to analyze URL requests. Specifically, authors analyzed tokenized URL requests with three-grams and transformed them into vectors based on likelihood ratio test. This method with Long Short-Term Memory (LSTM) model obtained 98.60% in accuracy. Saxe *et al*. [14] described a novel method for automatic analysis, which is to add an embedding layer in Convolutional Neural Networks (CNN). The optimal representation for URL requests will be generated through the training for the whole deep learning model. Compare with baseline models, this work performed better and achieved the highest accuracy at 99.3%.

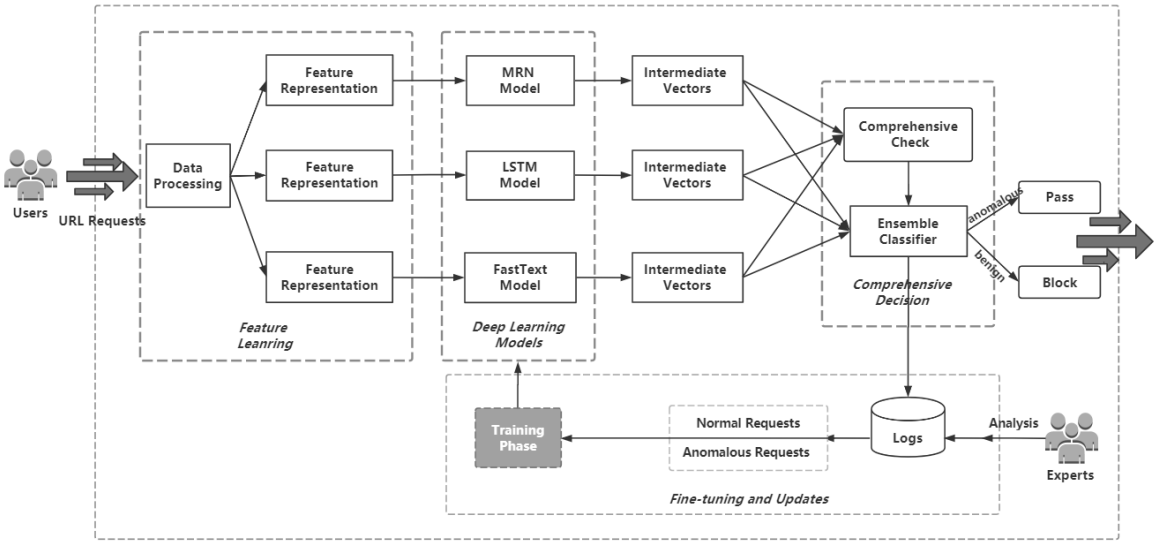

Fig. 1. The framework for the proposed EDL-WADS

Those previous researches based on three above-mentioned methods and deep learning show promising performance in detecting web attacks. Compared with other two methods, the third method based on automatic analysis show its superiority and becomes the focus of research direction in the future. As for deep learning models, systems proposed in existing studies mostly used single and simple models, such as SVM, LR, CNN, RNN, Naive Bayes and Random Forest while there is a risk that the system with single model may be bypassed of detecting specific attacks performed by hackers with new techniques studied in [24-25]. Hence, we conduct our research with automatically dissecting URL requests and utilizing three independent deep learning models for classification. To make full use of three deep learning model and improve detection performance, we utilized an ensemble classifier which can make comprehensive decision form multiple deep learning models.

III. Methodology

Our proposed Ensemble Deep Learning based Web Attack Detection System (EDL-WADS) consists of four modules: a). the feature learning module is applied to analyze URL requests and transform them into vectors with anomaly information attached; b). the deep learning models module is composed of three independent deep leaning models for classification; c). the comprehensive decision module is utilized to combine those parallel results in order to obtain the final results for detection; and d). the fine-tuning and updates module is designed to pretrain of updates classifiers. The framework of EDL-WADS is illustrated in Figure 1.

A. Feature Learning

Features are the core of all deep learning applications on account of deciding the ceiling of performance. As the first module of EDL-WADS, it plays a critical role in keeping the quality and integrity of the input data. Considering the diversity of URL requests, Data processing is utilized to remove unimportant information and decode the data flow. In the feature representation of EDL-WADS, we use two methods for URL analysis, which are a method based on embedding layers and an approach presented in [6, 26]. Significantly, we have concluded that automatic methods performed best in related works and utilized two automatic methods to analyze URL requests and transform them into vectors in EDL-WADS.

1) Method One for Feature Representation

One of the methods we utilized is proposed in previous work [26]. Most importantly, the URL requests will be tokenized by all punctuations and become easier and more readable and easier to handle after data processing. Firstly, we developed a dictionary, which consists of key words in HTML, JavaScript, SQL, Linux, Window, etc., and a table for transformation. Afterwards, we normalize the URL requests with retaining words in the dictionary and replace URL requests according to the transformation table. Specifically, the dictionary is showed in table I and the table for transformation is presented in Table II. For instance, the URL request tokenized by data processing looks like:

/ bbiq / users_home / userinformation . php ? union select 1 , 2 , (select load_file (' / var / www / html / sql-connections / db-creds . inc ')) - - +

It will be converted into a standard expression as:

/ bbiq / users_home / userinformation . php ? union select Numbers , Numbers , (select load_file (' / Purestring / Purestring / html / PureString / PureString . inc ')) - - +

To transform URL requests into vectors, we used CBOW and TF-IDF algorithms. Both are popular algorithms for text analysis in the field of Natural Language Processing (NLP). More precisely, every word and character are represented by a k-dimensional vector with semantic information attached. Likewise, each word or character is mapped to a k-dimensional vector with statistical analysis from TF-IDF algorithm. On account of the diversity of web attacks and URL requests, statistical and semantic features are both indispensable for this task. Additionally, we retain two types of vectors for every word and punctuation and obtain comprehensive feature vectors of them. Significantly, we utilize these feature vectors in different ways and design different classifiers, the details are explained in deep learning models section.

2) Method Two for Feature Representation

Another method for feature representation is based on embedding layer. Actually, embedding layer is a word2vec model which is added into a neural network for data transformation. In this work, we apply the same procedure for normalizing URL requests as described in method one for feature representation. The main structure of method two is the same as method one. These two methods have the same procedure for URL requests normalization and feature representing model. However, in method two, the embedding layer is added into deep learning models and is trained with classifiers, while in method one, the model for normalization and classifier are separate and model for normalization needs to be pretrained independently.

TABLE I
EXAMPLES OF KEYWORDS DEFINED

Description	Keywords
HTML	Doctype a abbr b big body br button caption center cite code color command dir dialog div font form frame head hr html iframe img input label link meta …
Javascript	Abstract arguments boolean break byte case catch char class* const continue debugger default delete do double else enum* eval export* import* in int …
SQL	Union select and or if order by limit concat create table column database insert update drop delete from index show set alert where having group between unique primary key ifnull…
Puctuation	All punctuations
…	…

TABLE II
TRANSPORTATION SCHEMA

Transformation	Description
SenString	Represent keywords which didn't show up in training data
Numbers	Represent numbers
UniString	Represent Unicode strings
MixString	Represent strings consist of characters a to z, '_', '-' and numbers
MD5String	Represent md5 strings
PureString	Represent strings consist of character a to z, '_'and '-'

B. Deep Learning Models

In EDL-WADS, the section of deep learning models is the key module for detecting web attacks. According to the feature vectors provided in the model of feature learning, we utilized three deep learning models for classification, they are MRN model, LSTM model and CNN model respectively. Particularly, there are two main reasons for we used three deep learning models instead of two or more deep learning models. Firstly, other models will be clearly affected if one model is compromised when there are two models. Secondly, more deep learning models will lead to more cost of computing source and time.

1) MRN Model

MRN is a new structure of computing unit, which has been improved on the bias of Residual Network (ResNet), proposed in previous work [26]. The structure of MRN is illustrated in Figure 2 and the equation of the unit is described as follows:

$$H(\mathbf{x}) = pool(\alpha F_1(\mathbf{x}) + \beta F_2(\mathbf{x}) + \gamma F_3(\mathbf{x})) \qquad (3\text{-}1)$$

Where α, β and γ are to be optimized with all parameters of the model in the training phase. In MRN, $F_2(\mathbf{x})$ and $F_3(\mathbf{x})$ are designed to analyze URL requests in a semantic way, and they are able to extract useful semantic features, $F_1(\mathbf{x})$ is a fast-track that retains all statistic information which includes information dropped by $F_2(\mathbf{x})$ and $F_3(\mathbf{x})$. The procedure of MRN is explained in Figure 3. We utilized a two-channel matrix for inputting into MRN model as researchers do for pictures in the field of computer version. As shown in Figure 3, the URL requests are represented by matrix in two channels, one is composed of CBOW vectors and the other is composed of TF-IDF vectors, so that EDL-WADS can effectively taking advantages of the MRN units.

The MRN model is designed with three parts, as illustrated in Figure 4: feature representation (Inputs), feature extraction and classification. The input is a two-channel matrix composed of semantic vectors and statistic vectors generated in the feature learning module. The part of feature extraction is composed of four parallel MRN layers that are referred as the structure of the "Inception". By using multiple MRN layers stacked, different scales of semantic and statistic features are increased while the depth of the model is still shallow. Through the concatenation and the flatten layer, all features from MRN layers will be

concatenated and flattened, then sent to classification module. The classifier is composed of three dense layers: full connected layer, batch normalization layer and sigmoid layer. There is a dropout layer after each dense layer which is omitted in figure 4.We provide all parameters in Table III.

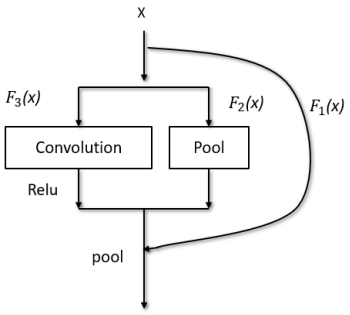

Fig. 2. The structure of MRN

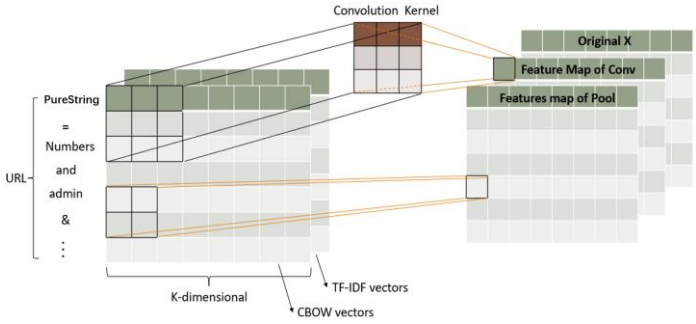

Fig. 3. The process of MRN

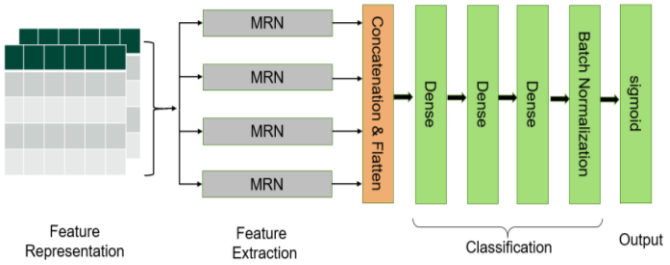

Fig. 4. The structure of MRN Model

TABLE III
PARAMETERS OF MRN MODEL

Kernel	C&Flatten	Dense	Dense	Dense
3×1, 5×1, 2×3, 3×5	128	64	32	2

2) LSTM Model

LSTM is the most widely used deep learning model in research on NLP. The URL requests are essentially texts. Therefore, it is common that we took the detection task as a text classification and designed a LSTM model for the task. The structure of LSTM model is shown in Figure 5 and the list of parameters are provided in Table IV. For LSTM model, the semantic and statistic vectors generated in method one of feature representation are utilized as input. We concatenate two types of k-dimensional vectors and used the combined 2k-dimension vectors as the input of the LSTM model. Specially, the LSTM model consists of LSTM layers, an MLP module and an output layer. The core of LSTM Model is the LSTM layers which is utilized to extract features from input vectors. The MLP model is used to map the output of LSTM layers and classify them. Finally, A sigmoid layer is designed to normalize the classification probability and make the final decision.

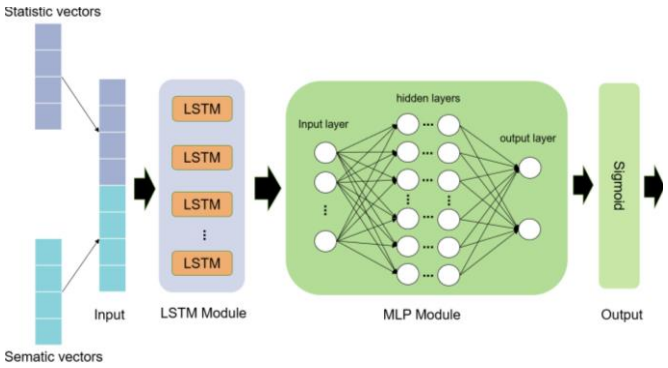

Fig. 5. The structure of LSTM Model

TABLE IV
PARAMETERS OF LSTM MODEL

LSTM hidden Units	MLP Input	MLP Hidden	MLP Hidden	MLP Output
128	128	48	24	2

3) CNN Model

In EDL-WADS, we designed a CNN model that uses a feature representation method based on the embedding layer. The structure of CNN model is illustrated in Figure 6 with parameters provided in Table V. The input of this CNN model is a sequential vector of normalized URL requests which processed by the approach described in method one of the feature representation. Embedding layer is utilized to convert these input words into vectors and propagate them to the CNN layers. Similarly, we used the same structure as MRN model to stack the CNN layers, and the three different CNN neural networks can increase the scale of the features. Concatenation and flatten layer will concatenate these features and propagate them into the classification model composed of two dense layers: a batch normalization layer and a sigmoid layer.

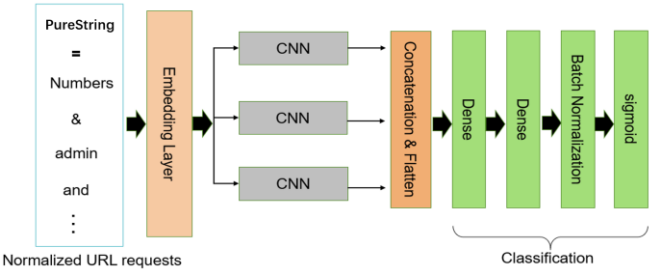

Fig. 6. The structure of CNN Model

TABLE V
PARAMETERS OF CNN MODEL

Embedding Layer	Kernels	C&Flatten	Dense	Dense
128	3×3, 3×5, 5×7	64	24	2

C. Comprehensive Decision

The comprehensive decision module is designed to combine those parallel results from multiple deep learning models and obtain the final decision for detection. Three deep learning models are used for classification, and each of them outputs an intermediate vector after its computing. To get the best predictive performance, we perform a comprehensive check and use an ensemble classifier. The comprehensive check is to calculate a vector V_r that denotes the reliability of results of every deep learning model, as described in Algorithm 1. First, we get V_m that represents the average of immediate vectors. Second, for each immediate vector V_i, a Euclidean distance between it and V_m is be calculated. Finally, we obtain a reliable vector V_r according to the Euclidean distance for every immediate vector V_i. The Euclidean distance shows the immediate vectors' reliability according to the normal fluctuation range of results of every model. Specifically, if the Euclidean distance is less than threshold ε, the immediate vector is considered as reliable and the value is then set to 1, or the immediate vector is considered as unreliable. Consequently, its value is set to 0.

In EDL-WADS, we used an MLP model as an ensemble classifier to combine all intermediate vectors and make the final decision. The structure of the ensemble classifier is depicted in Figure 7. The inputs of the model are vectors calculated using immediate vector V_i and reliability vector V_r. The concatenation and flatten layer will merge these vectors into one and propagate it to the MLP model. The MLP model and sigmoid layer will make the final decision on web attack detection.

Algorithm 1: Comprehensive Check

Input: Intermediate vectors from MRN model V_1, Intermediate vectors from LSTM model V_2, Intermediate vectors from CNN model V_3, The average of immediate vectors V_m, Thresholds ε.
Output: A vector V_r represnets the reliability of all intermediate vectors.

for each V_i, $i \in \{1,2,3\}$ **do**
  sum=0
  for V_i[k], k $\in dim(V_i)$ **do**
    dif=V_i[k]-V_m[i]
    sum=sum+dif^2
  end for
  if sum $\leq \varepsilon$ **then**
    $V_r[i]$=1
  else
    $V_r[i]$=0
  end if
end for

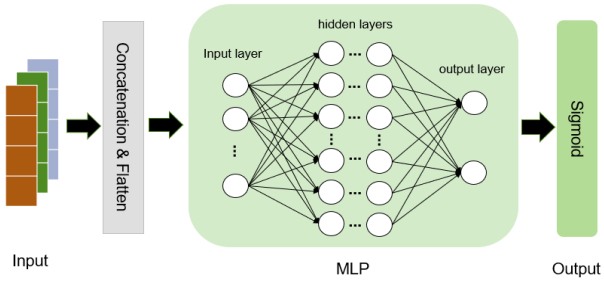

Fig. 7. The structure of ensemble classifier

D. Fine-tuning and Updates

Because of the complexity of real-world network environment and the diversity of web attacks, deep learning models in intrusion detection system (IDS) needs regular updates. As is shown in Figure 1, in order to improve the robustness and reliability of EDL-WADS, we integrate in it a feedback mechanism to fine-tune and update the system. In the fine-tuning and updates module, all raw URL requests, normalized data and detection results are recorded in a database to facilitate further analysis by the security experts. Moreover, EDL-WADS is designed to take advantage of experts' analysis to fine-tune deep learning models in the training phase and update these models incrementally in order to discover new web attacks. When one of the three models is being fine-tuned and updated, the remaining two other models continue to work. This ensures the fine-tuning and update on one model makes very little negative impact on the overall detection making. Most importantly, in terms of the reliability, our proposed system is fault tolerant, namely, when one deep learning model is under attack (e.g. attacks described in [24-25]), two other deep learning models are still active and making decisions jointly with very little performance degradation.

IV. Experimental Results and Discussion

To evaluate the proposed EDL-WADS, we conducted experiments on a synthetic dataset as a benchmark, a real-world dataset as well as a dataset collected in real-time by ourselves when performing attacks to the IoT network using attack tools such as sqlmap, Burpsuite, etc. As part of our experiments, we implemented EDL- WADS in a distributed environment and compared EDL-WADS with several approaches in the literature.

A. Datasets and Metrics

In order to evaluate EDL-WADS and compare it with existing approaches fairly, we used HTTP CSIC dataset 2010 (commonly referred to as CSIC 2010) [28] as a benchmark dataset. The CSIC 2010 dataset has been broadly used to evaluate IDS. It contains various of web attacks include SQL injection, cross-site scripting (XSS), buffer overflow, etc. Significantly, we extract 3329 SQL samples, 2053 XSS samples and 4812 benign samples and review them manually. Furthermore, we evaluate EDL-WADS on a real-word dataset which collected by a security company. There are 27614 SQL queries, 24834 XSS queries and 52448 benign queries in this dataset. Furthermore, the detection problem is served as a classification problem, and we calculate accuracy, True Positive Rate (TPR), False Positive Rate (FPR), precision using TP, TN, FP and FN defined in [26].

B. Experimental Results and Discussion

Firstly, we conduct experiments on MRN model. As shown in Figure 4, the structure of MRN model, four MRN layers are stacked to extract more semantic and statistic features. We set a value for every kernel based on experience. We then combine these four kernels into six groups. The details of these kernel combination groups are listed in table VI. In order to achieve the best group of kernels, we carried out experiments with six groups of kernels on CSIC 2010 dataset. The results are

summarized in Figure 8. More specifically, we first set group A based on our experiments and received promising results with accuracy, TPR and FPR all higher than 98.5%. We then make little changes from group A to group C, the performance increased slowly and achieved the highest in group C. However, the performances of accuracy and precision came to a sharp drop. We come to a conclusion that the kernel with size of 7×7 is too wide to extract useful features for MRN model. The accuracy and precision increased immediately when the kernel of 7×7 is replaced. In the feature representation, we map every word in the URL requests to a vector of k-dimension which is the row of the input matrix. The kernel of m×1 can extract static features of every word, so that group E comes last with no m×1 kernel in it while group C performs best with two m×1 kernels. Hence, we apply group C to the MRN model in our EDL-WADS.

TABLE VI
KERNELS GROUPS

Groups	Size of convolution kernels
Kernel-A	3×1, 2×2, 3×3, 5×5
Kernel-B	3×1, 2×3, 3×5, 5×5
Kernel-C	3×1, 5×1, 2×3, 3×5
Kernel-D	3×1, 3×3, 5×5, 7×7
Kernel-E	3×2, 3×3, 3×5, 5×5
Kernel-F	3×1, 3×3, 3×5, 5×5

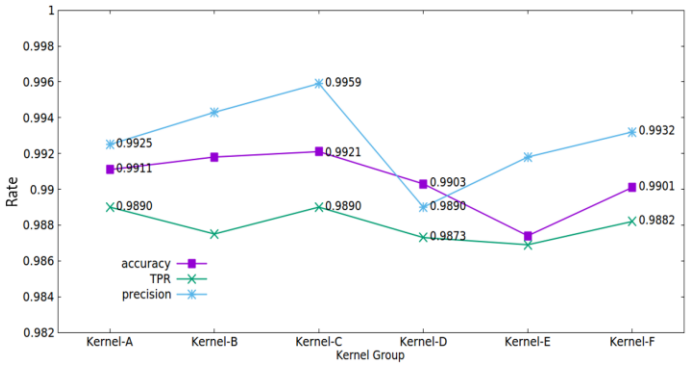

Fig. 8. Experimental results for kernels in MRN

Next, we implemented experiments for comparing EDL-WADS with existing approaches. In order to get a fair comparison, we conduct experiments on a benchmark dataset CSIC 2010. Particularly, three baseline models are tested. 1) a Specially Designed CNN for web attack detection (SDCNN) [27]. 2) a Web Application Firewall (WAF) using a Character-Level CNN (CLCNN) [15]. 3) A deep learning model consists of RNN and LSTM (RALM) [16]. The performances for comparison are listed in Table VII. RALM performs the best with a slight advantage than CLCNN. It achieves 98.56% at accuracy, 98.77% at TPR and 98.5% at precision. SDCNN comes last among the baseline models. All three models in EDL-WADS performed well, CNN models in EDL-WADS are very similar to CLCNN in all metrics. Actually, CNN model in EDL-WADS and CLCNN both used an embedding layer for feature representation. It seems that embedding layer is effective in representing URL requests. The results of LSTM model in EDL-WADS are slightly better than CNN. LSTM performs better than CNN in EDL-WADS, because the URL requests are mapped to vectors before being inputted in deep learning models in LSTM, while the mapping layer is embedded in deep learning model in CNN and may be influenced in the training phase. LSTM model in MRN achieves promising results. However, MRN shows better performance in all metrics. There may be two reasons: i) the MRN and LSTM in EDL-WADS have the same feature representations but different use-patterns of feature vectors, ii) LSTM model focuses on semantic analysis only, while MRN model can extract both semantic features and statistic features depending on its special structure. Besides, EDL-WADS system performs slightly higher than MRN and obtains the highest scores in accuracy, TPR and FPR. It demonstrates that the comprehensive check and ensemble classifier have the capability of combining results from multiple deep learning models accurately and comprehensively. As a result, it helped improve the detection performance of EDL-WADS.

TABLE VII
EXPERIMENTAL RESULTS ON CSIC 21010 DATASET

Model	ACC	TPR	FPR	Precision
RALM model with LSTM in [16]	0.9856	0.9877	0.0168	0.9850
SDCNN model in [27]	0.9649	0.9457	0.0135	0.9874
CLCNN model in [15]	0.9881	0.9853	0.0087	0.9921
MRN in EDL-WADS	**0.9921**	**0.9890**	**0.0046**	**0.9959**
LSTM in EDL-WADS	0.9901	0.9876	0.0071	0.9936
CNN in EDL-WADS	0.9876	0.9849	0.0094	0.9916
EDL-WADS	**0.9947**	**0.9929**	**0.0033**	**0.9970**

Furthermore, because of the limitations of the existing security dataset and the diversity of web attacks, public available datasets that are often used currently are not reliable enough to evaluate a web attack detection system. Further comparisons are carried out to evaluate the capacity of web attack detection ability of the proposed EDL-WADS. More specifically, we conducted experiments on a real-world dataset collected by a security company and the results are shown in Figure 9 and Figure 10. Compared with experimental results collected on CSIC 2020 dataset, the value of each metric of each approach is reduced. There are two main reasons according to our analysis. First, the CSIC 2010 dataset was generated in 2010, there are fewer types of web attacks than

TABLE VIII
TABLE PERFORMANCE OF WADS ON A REAL-TIME TRAFFIC

Malicious	Benign	TP	TN	FP	FN	ACC	TPR	Precision	FPR
6075	4360	6075	4358	2	**0**	**99.98%**	**100%**	**99.97%**	0.046%

today. Second, the CSIC 2010 dataset is synthetic and collected in the labs on a network environment that is much simpler than it in real-world. Therefore, the decrease in Figure 9 reflect the importance of using a real-world dataset for evaluating research results in the field of network security. As shown in Figure10, MRN performs the best among three individual models and CNN performs the worst. Finally, EDL-WADS outperforms all three individual models. It demonstrates that EDL-WADS is capable of combing MRN, LSTM and CNN models accurately. A comparison between EDL-WADS and existing approaches, which include DBPF, SDCNN and CLCNN, is carried out, EDL-WADS achieves superior performance with 99.17% in accuracy, 99.26% in TPR, 99.17% in Precision and 0.93% in FPR. The experimental results demonstrate that EDL-WADS performs better than existing works and can detect web attacks accurately with low false positives and negatives.

Eventually, another experiment is conducted to test how EDL-WADS performs in a real-world environment. For this purpose, we used a famous web application DVWA as a target and deployed EDL-WADS in a distributed environment to detect attacks against DVWA. Specifically, we take advantage of several security tools, which include sqlmap, burpsuite and XSStrike, to launch attacks against DVWA. The experimental results are illustrated in table VIII.

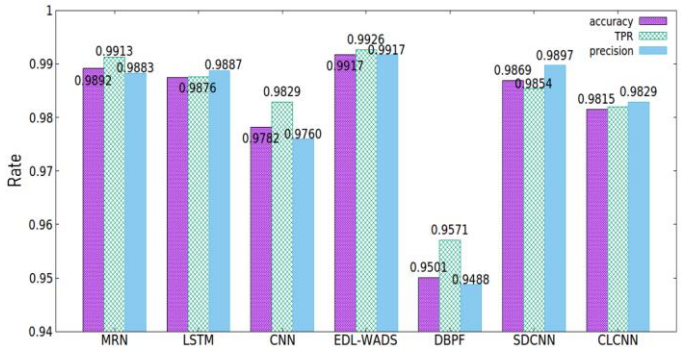

Fig. 9. Experimental results of ACC, TPR and Precision on a real-world dataset

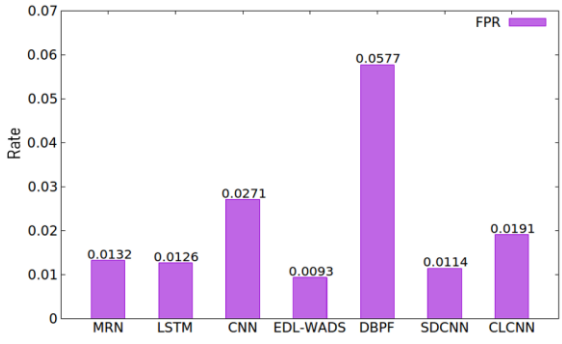

Fig. 10. Experimental results for kernels in MRN

The results obtained show that EDL-WADS achieves the highest accuracy, TPR and precision as well as the lowest FPR. In this experiment, we collected 6075 anomalous requests from security tools and 4360 normal requests by programs automatically. The EDL-WADS system achieved 100% in TPR, which demonstrates that all web attacks are detected accurately. The other metrics also demonstrated high values. Only two of all requests are detected wrongly: normal requests were detected as malicious ones. The results seem to be unexpectedly ideal. After several rounds of analysis, we found out the reason: these security tools that we have used to perform attacks all use common and simple security rules to scan the target system. EDL-WADS detects such simple and common attacks with very high accuracy. Nonetheless, the EDL-WADS truly demonstrated its effectiveness on real-time web attacks detection given these attack tools that we have selected are the most commonly used ones on internet.

V. Conclusion

In this work, we proposed a novel Web Attack Detection System, EDL-WADS, for IoTs. Specifically, the EDL-WADS consists of four modules: 1) A feature learning module for URL request representations. 2) A deep learning module composed of three deep learning models for producing different representations of URL requests in order to exploit the advantages from a variety of classification. 3) A comprehensive decision module for combing the results from the three deep learning models and making the final decision with an ensemble classifier. 4) A fine-tuning and updates module for fine-tuning and updating the three deep learning models in real time.

To evaluate the proposed EDL-WADS, we have carried out experiments on different datasets. The experimental results on a benchmark dataset CSIC 2010 show that EDL-WADS outperforms all selected baseline models. The overall performance was 99.47% on accuracy, 99.29% on true positive rate and 99.70% on precision, with a low false positive rate of 0.0033. Furthermore, experiments were carried out on a real-world dataset. The results confirm that EDL-WADS has a superior performance compared to several existing approaches. However, there are two primary limitations that require further improvement in the future: 1. the current EDL-WADS system can only detect SQL injection and cross-site scripting attacks. 2. the CNN model in EDL-WADS does not perform as well as we had expected, therefore a more desirable model should replace it in the future. Thus, our future research direction will focus on improving the EDL-WADS for detecting additional types of web attacks (e.g., command injection and file inclusion) and exploring alternative deep learning models to better the performance of the current system.

VI. Reference

[1] M. Lin, C. Chiu, Y. Lee, et al. Malicious URL filtering - A big data application. 2013 *IEEE international conference on big data*. IEEE, 2013, pp. 589-596.

[2] D. Kar, S. Panigrahi, S. Sundararajan, et al. SQLiDDS: SQL injection detection using query transformation and document similarity. *International Conference on Distributed Computing and Internet Technology*. Springer, 2015, pp. 377-390.

[3] A. Le, A. Markopoulou, M. Faloutsos, et al. Phishdef: Url names say it all. 2011 Proceedings IEEE INFOCOM. IEEE, 2011, pp. 191-195.

[4] J. Qiu, L. Du, D. Zhang, et al. Nei-TTE: Intelligent Traffic Time Estimation Based on Fine-grained Time Derivation of Road Segments for Smart City. *IEEE Transactions on Industrial Informatics*. vol. 16, no. 4, pp. 2659-2666, 2020.

[5] P. Bisht, P. Madhusudan, V. N. Venkatakrishnan, et al. Dynamic candidate evaluations for automatic prevention of SQL injection attacks. *ACM Trans. Inf. Syst. Security*, vol. 13, no. 2, 2010.

[6] C. Luo, S. Su, Y. Sun, et al. A Convolution-Based System For Malicious

URL Requests Detection. *Computers Materials & Continua (CMC)*. vol. 61, no. 3, pp. 399-411, 2019.

[7] M. Li, Y. Sun, H. Lu, et al. Deep Reinforcement Learning for Partially Observable Data Poisoning Attack in Crowdsensing Systems. *IEEE Internet of Things Journal*. vol. 7, no. 7, pp. 6266-6278, 2020.

[8] Y. H. Hwang. "IoT Security & Privacy: Threats and Challenges.". *Acm Workshop on Iot Privacy*. ACM, 2015, pp. 1-1.

[9] A. Jamdagni, Z. Tan, X. He, et al. RePIDS: A multi-tier Real-time Payload-based Intrusion Detection System. *Computer networks*, vol. 57, no. 3, pp. 811-824, 2013.

[10] Z. Tan, A. Jamdagni, X. He, et al. A System for Denial-of-Service Attack Detection Based on Multivariate Correlation Analysis. *IEEE Transactions on Parallel & Distributed Systems*, vol. 25, no. 2, pp. 447-456, 2014.

[11] C. Torrano-Gimenez, H. T. Nguyen, G. Alvarez, et al. Applying feature selection to payload-based web application firewalls. 2011 *Third International Workshop on Security and Communication Networks (IWSCN)*. IEEE, 2011, pp. 75-81.

[12] J. Macqueen. Some Methods for Classification and Analysis of MultiVariate Observations. *Proc of Berkeley Symposium on Mathematical Statistics & Probability*. vol. 1, no. 14, pp. 281-297, 1965.

[13] D. Kar, S. Panigrahi, S. Sundararajan, et al. SQLiGoT: Detecting SQL injection attacks using graph of tokens and SVM. *Computers & Security*, vol. 60, pp. 206-225, 2016.

[14] J. Saxe and K. Berlin. eXpose: A character-level convolutional neural network with embeddings for detecting malicious URLs, file paths and registry keys. *arXiv preprint arXiv:1702.08568*, 2017.

[15] M. Ito and H. Iyatomi. Web application firewall using character-level convolutional neural network. 2018 *IEEE 14th International Colloquium on Signal Processing & Its Applications (CSPA)*. IEEE, 2018, pp. 103-106.

[16] J. Liang, W. Zhao and W. Ye. Anomaly-based web attack detection: a deep learning approach. *Proceedings of the 2017 VI International Conference on Network, Communication and Computing*. 2017, pp. 80-85.

[17] J. Qiu, Z. Tian, C. Du, et al. A Survey on Access Control in the Age of Internet of Things. *IEEE Internet of Things Journal*. vol. 7, no. 6, pp. 4682-4696, 2020.

[18] J. Ma, L. K. Saul, S. Savage, et al. Beyond blacklists: learning to detect malicious Web sites from suspicious URLs. *Acm Sigkdd International Conference on Knowledge Discovery & Data Mining*. ACM, 2009, pp. 1245-1254.

[19] I. Lee, S. Jeong, S. Yeo, et al. A novel method for SQL injection attack detection based on removing SQL query attribute values. *Mathematical & Computer Modelling*, vol.55, no.1-2, pp. 58-68, 2012.

[20] F. Yong, P. Jiayi, L. Liang, and H. Cheng. WOVSQLI: Detection of SQL Injection Behaviors Using Word Vector and LSTM. *In Proceedings of the 2nd International Conference on Cryptography, Security and Privacy (ICCSP 2018)*. ACM, 2018, pp. 170–174.

[21] T. Liu, Y. Qi, L. Shi and J. Yan. Locate-Then-Detect: Real-time Web Attack Detection via Attention-based Deep Neural Networks. *Joint Conference on Artificial Intelligence (JCAI)*, 2019, pp. 4725-4731.

[22] Y. Zhou and G. Cheng. An Efficient Intrusion Detection System Based on Feature Selection and Ensemble Classifier. *arXiv: Cryptography and Security*, 2019.

[23] R. Vinayakumar, K. P. Soman and P. Poornachandran. Detecting malicious domain names using deep learning approaches at scale. *Journal of Intelligent & Fuzzy System*. vol. 34, no.3, pp. 1355-1367, 2018.

[24] M. E. Ahmed and K. Hyoungshick. "Poster: Adversarial Examples for Classifiers in High-Dimensional Network Data," *in ACM SIGSAC Conference on Computer and Communications Security*, ACM, 2017, pp. 2467-2469.

[25] N. Papernot, P. Mcdaniel, I. Goodfellow, et al. "Practical black-box attacks against machine learning," *in ACM on Asia Conference on Computer and Communications Security*, ACM, 2017, pp. 506-519.

[26] Z. Tian, C. Luo, J. Qiu, et al, "A Distributed Deep Learning System for Web Attack Detection on Edge Devices," *in IEEE Transactions on Industrial Informatics*, vol. 16, no. 3, 2020.

[27] M. Zhang, B. Xu, S. Bai, et al. A deep learning method to detect web attacks using a specially designed cnn. *International Conference on Neural Information Processing*. Springer, Cham, 2017, pp. 828-836.

[28] C. T. Giménez, A. P. Villegas and G. Á. Marañón. HTTP data set CSIC 2010. *Information Security Institute of CSIC* (Spanish Research National Council), 2010.