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

L.C.Usha Maheswari[1], G. Srivalli[2], G . Shivani[3], G. Sai Nikitha[4], K.Kaveri[5]
[1]Assistant Professor, Department of CSE(CS), MallaReddy Engineering College for Women, Hyderabad, TS, India.
Email: maheswari07usha@gmail.com
[2,3,4,5]UG Students, Department of CSE(CS), MallaReddy Engineering College for Women, Hyderabad, TS, India.

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 article, 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.

INTRODUCTION
AS ONE of the fastest-growing and widely used technologies on the Internet, Internet of Things (IoT) extends the Manuscript received June 29, 2020; revised August 30, 2020, October 7, 2020, and October 30, 2020; accepted November 13, 2020. Date of publication November 17, 2020; date of current version May 3, 2021. This work 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, and in part by the Guangdong Province Key Research and Development Plan under Grant 2019B010137004. Paper no. TII-20-3138. *(Correspondin author: Zhihong Tian.* Chaochao Luo and Jie Gan are with the ADLab of Venustech, Beijing 100001, China (e-mail: luochaochaoforaffair@gmail.com; charles2gan@gmail.com). Zhiyuan Tan is with the School of Computing, Edinburgh Napier University, EH10 5DT Edinburgh, U.K. (e-mail: z.tan@napier.ac.uk). Geyong Min is with the Department of Computer Science, University of Exeter, EX4 4QF Exeter, U.K. (e-mail: g.min@exeter.ac.uk).Wei Shi is with the School of Information Technology, Carleton University,Ottawa, ON K1S 5B6, Canada (e-mail: weishi@cunet.carleton.ca).

Zhihong Tian is with theCyberspace Institute of Advanced Technology, Guangzhou University, Guangzhou 510006, China (e-mail: tianzhihong@gzhu.edu.cn). Color versions of one or more of the figures in this article are available online at https://doi.org/10.1109/TII.2020.3038761 Digital Object Identifier 10.1109/TII.2020.3038761 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 an 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].Recentworks covering web attack detection systems (WADS) 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 article, we propose a novel 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. 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.

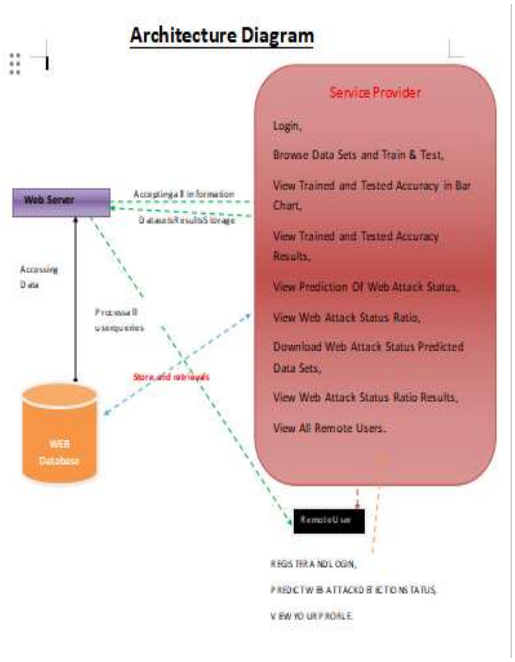
The contributions of this article are summarized as follows

1) We propose EDL-WADS, a novel ensemble deep learning- based system that can detect anomalous que ies in which malicious codes are attached in an IoT network.

2) 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.

3) An ensemble classifier is utilized in EDL-WADS to improve the detection performance by combining results from different classifiers based on multilayer perceptrons (M P).4) 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 article is organized as follows. In Section II, we present the state-of-the-art works in this research field. Section III introduces the proposed system. In Section IV, we present the experimental results and their analysis. Finally,Section V concludesthis article.



Existing System

Ma *et al.* [18] used static features and evaluated the methods with the 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 make 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. [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 performance in web attack detection. Compared with the frmer method, the second method reduces manual intervention to some extent and overcomes the disadvantage of the first method. However, the second method does not show significant improvement in the performance of web attack detection.

As for the third method, it has been the 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 the likelihood ratio test. This method with the long short-term memory (LSTM) model obtained 98.60% in accuracy. Saxe and Berlin [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. Compared with baseline models, this work performed better and achieved the highest accuracy at 99.3%.

Disadvantages

1) 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.

2) 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.

3) The method of analyzing URL requests and transforming them into vectors automatically shows its superior capability of representing URL requests accurately. It has become the state-of-the-art method in the field of web attack detection.

Proposed System

In this article, we propose a novel WADS for IoT networks, based on ensemble deep learning. Specifically, the proposed EDL-WADStakes 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. 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 article are summarized as follows.

1) The system proposes EDL-WADS, a novel ensemble deep learning- based system that can detect anomalous queries in which malicious codes are attached in an IoT network.

2) The system utilizes a group of deep learning models to produce different representations of URL requests in order to exploit the advantages from a variety of classification.

3) An ensemble classifier is utilized in EDL-WADS to improve the detection performance by combining results from different classifiers based on multilayer perceptrons (MLP).

4) The Proposed system compares 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

Advantages

1) The feature learning module is applied to analyze URL requests and transform them into vectors with anomaly information attached.

2) The deep learning models module is composed of three independent deep learning models for classification.

3) The comprehensive decision module is utilized to combine those parallel results in order to obtain the final results for detection.

4) The fine-tuning and updates module is designed to pretrain updates classifiers. The framework of EDL-WADS is illustrated in the proposed system in effective way.

Decision tree classifiers

Decision tree classifiers are used successfully in many diverse areas. Their most important feature is the capability of capturing descriptive decision making knowledge from the supplied data. Decision tree can be generated from training sets. The procedure for such generation based on the set of objects (S), each belonging to one of the classes C1, C2, …, Ck is as follows:

Step 1. If all the objects in S belong to the same class, for example Ci, the decision tree for S consists of a leaf labeled with this class
Step 2. Otherwise, let T be some test with possible outcomes O1, O2,…, On. Each object in S has one outcome for T so the test partitions S into subsets S1, S2,… Sn where each object in Si has outcome Oi for T. T becomes the root of the decision tree and for each outcome Oi we build a subsidiary decision tree by invoking the same procedure recursively on the set Si.

Gradient boosting

Gradient boosting is a machine learning technique used in regression and classification tasks, among others. It gives a prediction model in the form of an ensemble of weak prediction models, which are typically decision trees.[1][2] When a decision tree is the weak learner, the resulting algorithm is called gradient-boosted trees; it usually outperforms random forest.A gradient-boosted trees model is built in a stage-wise fashion as in other boosting methods, but it generalizes the other methods by allowing optimization of an arbitrary differentiable loss function.

K-Nearest Neighbors (KNN)

- **Simple, but a very powerful classification algorithm**
- **Classifies based on a similarity measure**
- **Non-parametric**
- **Lazy learning**
- **Does not "learn" until the test example is given**
- **Whenever we have a new data to classify, we find its K-nearest neighbors from the training data**

Example

- **Training dataset consists of k-closest examples in feature space**
- **Feature space means, space with categorization variables (non-metric variables)**
- **Learning based on instances, and thus also works lazily because instance close to the input vector for test or prediction may take time to occur in the training dataset**

Logistic regression Classifiers

Logistic regression analysis studies the association between a categorical dependent variable and a set of independent (explanatory) variables. The name *logistic regression* is used when the dependent variable has only two values, such as 0 and 1 or Yes and No. The name *multinomial logistic regression* is usually reserved for the case when the dependent variable has three or more unique values, such as Married, Single, Divorced, or Widowed. Although the type of data used for the dependent variable is different from that of multiple regression, the practical use of the procedure is similar.

Logistic regression competes with discriminant analysis as a method for analyzing categorical-response variables. Many statisticians feel that logistic regression is more versatile and better suited for modeling most situations than is discriminant analysis. This is because logistic regression does not assume that the independent variables are normally distributed, as discriminant analysis does.

This program computes binary logistic regression and multinomial logistic regression on both numeric and categorical independent variables. It reports on the regression equation as well as the goodness of fit, odds ratios, confidence limits, likelihood, and deviance. It performs a comprehensive residual analysis including diagnostic residual reports and plots.

It can perform an independent variable subset selection search, looking for the best regression model with the fewest independent variables. It provides confidence intervals on predicted values and provides ROC curves to help determine the best cutoff point for classification. It allows you to validate your results by automatically classifying rows that are not used during the analysis.

Naïve Bayes

The naive bayes approach is a supervised learning method which is based on a simplistic hypothesis: it assumes that the presence (or absence) of a particular feature of a class is unrelated to the presence (or absence) of any other feature .
Yet, despite this, it appears robust and efficient. Its performance is comparable to other supervised learning techniques. Various reasons have been advanced in the literature. In this tutorial, we highlight an explanation based on the representation bias. The naive bayes classifier is a linear classifier, as well as linear discriminant analysis, logistic regression or linear SVM (support vector machine). The difference lies on the method of estimating the parameters of the classifier (the learning bias).

While the Naive Bayes classifier is widely used in the research world, it is not widespread among practitioners which want to obtain usable results. On the one hand, the researchers found especially it is very easy to program and implement it, its parameters are easy to estimate, learning is very fast even on very large databases, its accuracy is reasonably good in comparison to the other approaches. On the other hand, the final users do not obtain a model easy to interpret and deploy, they does not understand the interest of such a technique.

Thus, we introduce in a new presentation of the results of the learning process. The classifier is easier to understand, and its deployment is also made easier. In the first part of this tutorial, we present some theoretical aspects of the naive bayes classifier. Then, we implement the approach on a dataset with Tanagra. We compare the obtained results (the parameters of the model) to those obtained with other linear approaches such as the logistic regression, the linear discriminant analysis and the linear SVM. We note that the results are highly consistent. This largely explains the good performance of the method in comparison to others. In the second part, we use various tools on the same dataset (**Weka 3.6.0**, **R 2.9.2**, **Knime 2.1.1**, **Orange 2.0b** and **RapidMiner 4.6.0)**. We try above all to understand the obtained results.

Random Forest

Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks that operates by constructing a multitude of decision trees at training time. For classification tasks, the output of the random forest is the class selected by most trees. For regression tasks, the mean or average prediction of the individual trees is returned. Random decision forests correct for decision trees' habit of overfitting to their training set. Random forests generally outperform decision trees, but their accuracy is lower than gradient boosted trees. However, data characteristics can affect their performance.
The first algorithm for random decision forests was created in 1995 by Tin Kam Ho[1] using

the random subspace method, which, in Ho's formulation, is a way to implement the "stochastic discrimination" approach to classification proposed by Eugene Kleinberg.
An extension of the algorithm was developed by Leo Breiman and Adele Cutler, who registered "Random Forests" as a trademark in 2006 (as of 2019, owned by Minitab, Inc.).The extension combines Breiman's "bagging" idea and random selection of features, introduced first by Ho[1] and later independently by Amit and Geman[13] in order to construct a collection of decision trees with controlled variance.
Random forests are frequently used as "blackbox" models in businesses, as they generate reasonable predictions across a wide range of data while requiring little configuration.

SVM

In classification tasks a discriminant machine learning technique aims at finding, based on an *independent and identically distributed* (*iid*) training dataset, a discriminant function that can correctly predict labels for newly acquired instances. Unlike generative machine learning approaches, which require computations of conditional probability distributions, a discriminant classification function takes a data point *x* and assigns it to one of the different classes that are a part of the classification task. Less powerful than generative approaches, which are mostly used when prediction involves outlier detection, discriminant approaches require fewer computational resources and less training data, especially for a multidimensional feature space and when only posterior probabilities are needed. From a geometric perspective, learning a classifier is equivalent to finding the equation for a multidimensional surface that best separates the different classes in the feature space.

SVM is a discriminant technique, and, because it solves the convex optimization problem analytically, it always returns the same optimal hyperplane parameter—in contrast to *genetic algorithms* (*GAs*) or *perceptrons*, both of which are widely used for classification in machine learning. For perceptrons, solutions are highly dependent on the initialization and termination criteria. For a specific kernel that transforms the data from the input space to the feature space, training returns uniquely defined SVM model parameters for a given training set, whereas the perceptron and GA classifier models are different each time training is initialized. The aim of GAs and perceptrons is only to minimize error during training, which will translate into several hyperplanes' meeting this requirement.

Modules

Service Provider

In this module, the Service Provider has to login by using valid user name and password. After login successful he can do some operations such as . Login, Browse Data Sets and Train & Test, View Trained and Tested Accuracy in Bar Chart, View Trained and Tested Accuracy Results, View Prediction Of Web Attack Status, View Web Attack Status Ratio,

Download Web Attack Status Predicted Data Sets, View Web Attack Status Ratio Results, View All Remote Users.

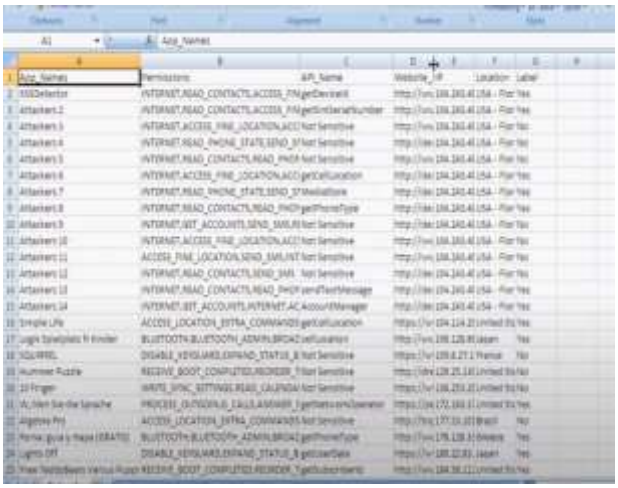
View and Authorize Users

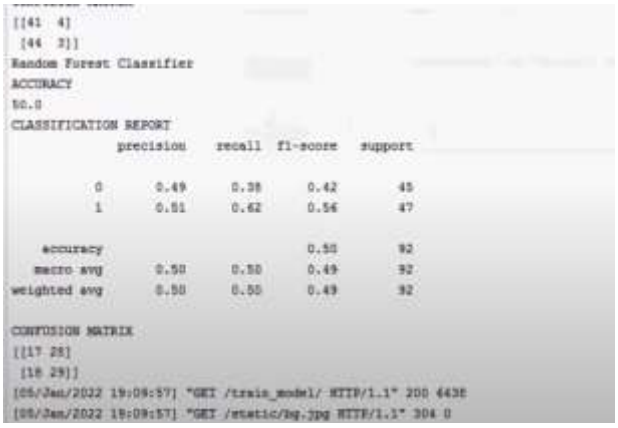
In this module, the admin can view the list of users who all registered. In this, the admin can view the user's details such as, user name, email, address and admin authorizes the users.

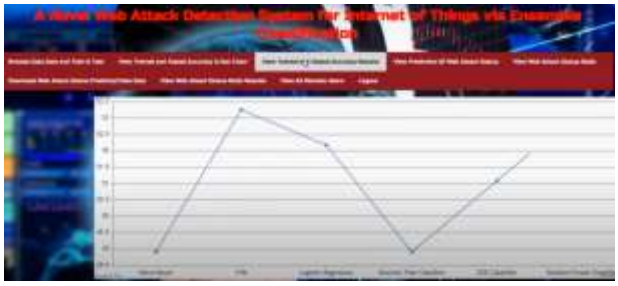
Remote User

In this module, there are n numbers of users are present. User should register before doing any operations. Once user registers, their details will be stored to the database. After registration successful, he has to login by using authorized user name and password. Once Login is successful user will do some operations like REGISTER AND LOGIN, PREDICT WEB ATTACK DETECTION STATUS, VIEW YOUR PROFILE.

Results







CONCLUSION

In this article, we proposed a novel WADS, EDL-WADS, for IoTs. Specifically, the EDL-WADSconsisted 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 carried out experiments on different datasets. The experimental results on a benchmark dataset CSIC 2010 showed that EDL-WADS outperforms all selected baseline models. The overall performance was 99.47% on accuracy, 99.29% on TPR, and 99.70% on precision, with a low FPR of 0.0033. Furthermore, experiments were carried out on a real-world dataset. The results confirmed that EDL-WADS have a superior performance compared to several existing approaches. However, there were two primary limitations that required 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.

REFERENCES

[1] M. Lin, C. Chiu, Y. Lee, and H. Pao, "Malicious URL filtering—A big data application," in *Proc. IEEE Int. Conf. Big Data*, 2013, pp. 589–596.

[2] D. Kar, S. Panigrahi, and S. Sundararajan, "SQLiDDS: SQL injection detection using query transformation and document similarity," in *Proc. Int. Conf. Distrib. Comput. Internet Technol.*, 2015, pp. 377–390.

[3] A. Le, A. Markopoulou, and M. Faloutsos, "PhishDef: URL names say it all," in *Proc. IEEE INFOCOM*, 2011, pp. 191–195.

[4] J. Qiu, L. Du, D. Zhang, S. Su, and Z. Tian, "Nei-TTE: Intelligent traffic time estimation based on fine-grained time derivation of road segments for smart city," *IEEE Trans. Ind. Informat.*, vol. 16, no. 4, pp. 2659–2666, Apr. 2020.

[5] P. Bisht, P.Madhusudan, and V. N. Venkatakrishnan, "Dynamic candidate evaluations for automatic prevention of SQL injection attacks," *ACM Trans. Inf. Syst. Secur.*, vol. 13, no. 2, pp. 398–404, 2010.

[6] C. Luo, S. Su, andY. Sun, "A convolution-based system formaliciousURL

requests detection," *Comput.Mater. Continua*, vol. 61, no. 3, pp. 399–411, 2019.
[7] M. Li, Y. Sun, H. Lu, S. Maharjan, and Z. Tian, "Deep reinforcement learning for partially observable data poisoning attack in crowdsensing systems," *IEEE Internet Things J.*, vol. 7, no. 7, pp. 6266–6278, Jul. 2020.
[8] Y. H. Hwang, "IoT security & privacy: Threats and challenges," in *Proc. 1st Acm Workshop on Iot Privacy Trust and Security*, 2015, p. 1.
[9] A. Jamdagni, Z. Tan, and X. He, "RePIDS: A multi-tier real-time payload-based intrusion detection system," *Comput. Netw.*, vol. 57, no. 3, pp. 811–824, 2013.
[10] Z. Tan, A. Jamdagni, X. He, P. Nanda, and R. P. Liu, "A system for denialof-service attack detection based on multivariate correlation analysis," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 2, pp. 447–456, Feb. 2014.
[11] C.Torrano-Gimenez, H.T. Nguyen, G. Alvarez, S. Petrovi´c, andK. Franke, "Applying feature selection to payload-based web application firewalls," in *Proc. 3rd Int. Workshop Secur. Commun. Netw.*, 2011, pp. 75–81.
[12] J. Macqueen, "Some methods for classification and analysis ofmultivariate observations," in *Proc. 5th Berkeley Symp. Math. Statist. Probability*, 1965, vol. 1, no. 14, pp. 281–297.
[13] D. Kar, S. Panigrahi, and S. Sundararajan, "SQLiGoT: Detecting SQL injection attacks using graph of tokens and SVM," *Comput. Secur.*, 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," 2017, *arXiv:1702.08568*.
[15] M. Ito and H. Iyatomi, "Web application firewall using character-level convolutional neural network," in *Proc. IEEE 14th Int. Colloq. Signal Process. Its Appl.*, 2018, pp. 103–106.
[16] J. Liang, W. Zhao, and W. Ye, "Anomaly-based web attack detection: A deep learning approach," in *Proc. VI Int. Conf. Netw., Commun. Comput.*, 2017, pp. 80–85.
[17] J. Qiu, Z. Tian, and C. Du, "A survey on access control in the age of Internet of things," *IEEE Internet Things J.*, vol. 7, no. 6, pp. 4682–4696, Jun. 2020.
[18] J. Ma, L. K. Saul, and S. Savage, "Beyond blacklists: Learning to detect malicious web sites from suspicious URLs," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2009, pp. 1245–1254.
[19] I. Lee, S. Jeong, and S. Yeo, "A novel method for SQL injection attack detection based on removing SQL query attribute values," *Math. Comput. 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 *Proc. 2nd Int. Conf. Cryptography, Secur. Privacy*, 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," in *Proc. Joint Conf.*

Artif. Intell., 2019, pp. 4725–4731.
[22] Y. Zhou and G. Cheng, "An efficient intrusion detection system based on feature selection ensemble classifier," *Computer Networks.*, vol. 174, 2020, Art. no. 107247.
[23] R. Vinayakumar, K. P. Soman, and P. Poornachandran, "Detecting malicious domain names using deep learning approaches at scale," *J. Intell. Fuzzy Syst.*, 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 *Proc. ACMSIGSACConf. Comput. Commun. Secur.*, 2017, pp. 2467–2469.
[25] N. Papernot, P.Mcdaniel, and I. Goodfellow, "Practical black-box attacks against machine learning," in *Proc. ACM Asia Conf. Comput. Commun. Secur.*, 2017, pp. 506–519.