Communication Papers of the of the 17th Conference on Computer Science and Intelligence Systems pp. 269–274
DOI: 10.15439/2022F147
ISSN 2300-5963 ACSIS, Vol. 32

Web Intrusion Detection Using Character Level Machine Learning Approaches with Upsampled Data

1st Talya Tümer Sivri
Defense and Information Systems
BITES
Ankara, TURKEY
talya.tumer@bites.com.tr

2nd Nergis Pervan Akman
Defense and Information Systems
BITES
Ankara, TURKEY
nergis.pervan@bites.com.tr

3rd Ali Berkol
Defense and Information Systems
BITES
Ankara, TURKEY
ali.berkol@bites.com.tr

4th Can PEKER
Training Tech. and Digital Platforms
BITES
Ankara, TURKEY
can.peker@bites.com.tr

***Abstract*—Today, people fulfill their needs in many areas such as shopping, health, and finance online. Besides many well-meaning people who use websites for their own needs, there are also people who send attack requests to get these people's personal data, get website owners' information, and damage the application. The attack types such as SQL injection and XSS can seriously harm web applications and users. Detecting these cyber-attacks manually is very time-consuming and difficult to adapt to new attack types. Our proposed study performs attack detection using different machine learning and deep learning approaches with a larger dataset obtained by combining CSIC 2012 and ECML/PKDD datasets. In this study, we evaluated our classification results which experimented with different algorithms based on computation time and accuracy. In addition to applying different algorithms, experiments on various learning models were applied with our data upsample method for balancing the dataset labels. As a result of the binary classification, LSTM achieves the best result in terms of accuracy, and a positive effect of the upsampled data on accuracy has been observed. LightGBM was the algorithm with the highest performance in terms of computation time.**

***Index Terms*—Web Application Firewall, Intrusion Detection, Data Upsampling, CNN, LSTM, XGBoost, LightGBM, Random Forest, Deep Learning**

I. Introduction

With the increase of devices and applications such as the Internet of Things (IoT), efforts to ensure the security of these applications are also increasing. Ensuring the security of web applications is a critical step for the continuity of the application. The Verizon 2020 Data Breach Investigations Report [5] found that web application attacks doubled in 2019, to nearly 43% of all attacks analyzed in the report. It is still an actively used approach to detect and prevent Hyper Text Transfer Protocol (HTTP) requests from coming into the application with rule-based software. However, in rule-based software, attack detection depends on the rule. It is not adapted for incoming attacks outside the rules. Apart from detecting attacks with the signature-based software, there are studies in the literature that classify using traditional machine learning [1], [2] and deep learning algorithms [4], [18], [23].

Along with detecting incoming attacks with high accuracy, rapid detection is also important for quick intervention [3]. For this reason, machine learning algorithms used in this study, which take less training time and detection time and need less data, have great importance in attack detection.

Hoang et al. [8] have included Naive Bayes, Support Vector Machine (SVM), Decision Tree, and Random Forest (RF) for binary classification. The best performance is obtained with RF, and the worst is Naive Bayes. The highest performance with 99.68% accuracy is obtained with RF.

The use of Natural Language Processing (NLP) techniques is common due to the frequent use of meaningful or meaningless characters in extracting the attributes of requests coming to the application. Since character-level Convolutional Neural Network (CNN) is constructed using vectors of a limited character string, it is very advantageous in terms of computational costs [4]. They build traditional CNN with two convolutional layers and also build CNN architecture which is merged final layers in parallel. They proposed a faster and less computational approach owing to character representation.

As a different feature extraction approach, the vectors of words are extracted and classified by deep learning. Zhang [3] et al. split HTTP requests according to certain characters and extract words and generate word vectors using Word2Vec, which is a word embedding algorithm. Then they calculated the Term Frequency — Inverse Document Frequency (TF-

IDF) value from the word vectors for dimension reduction. One of the two algorithms used in this study, Light Gradient Boosting Machine (LightGBM), has lower accuracy than another algorithm, Category Boosting (CatBoost), using three available datasets, namely, HTTP DATASET CSIC 2010, UNSW-NB15, and malicious Uniform Resource Locators (URL). Different word embedding algorithms such as Global Vectors (Glove) for Word Representation and Fasttext are compared with accuracy, true-positive rate, and false-positive rate performance metrics.

With the increase in the use of machine learning algorithms, the need for data increases, and the importance of approaches such as systems that need less data, data augmentation studies, and automatic data generation are increasing at the same level. Therefore, machine learning-based studies are now moving towards high-quality data augmentation [20]–[22] of data or producing an effective model with fewer data.

Both speed and accuracy are of great importance in intrusion detection. In this study, we aimed to demonstrate this evaluation by conducting experiments on different algorithms. In addition, one of the biggest aims of this study is to increase the data by diversifying the different headers in the data. In this way, we have generalized the headers that appear only in attacks or only in normal requests.

The main contributions of this study are as follows: i) In a binary classification of HTTP requests, both time and accuracy performances of traditional machine learning algorithms and deep learning algorithms are compared. ii) The effect of increasing the dataset size using upsampling approaches on the obtained model accuracies is observed.

II. Related Work

Intrusion detection is vital for vulnerable machines and applications by enabling the detection and subsequent prevention of malicious software or activity. Before the anomaly detection step algorithmically, feature selection and extraction [10]–[14] of HTTP requests is an important step that affects performance. In addition to these traditional feature analysis studies, there are also information extraction studies using NLP techniques. Liu et al. [15] developed a system that works together with the optimized SVM algorithm and feature analysis work. HTTP requests were analyzed and selected using expert knowledge. SVM is used as a classification algorithm, and a grid search algorithm is used for parameter optimization. The results of this study, using the CSIC 2010 dataset, can detect attacks with high accuracy performance.

Providing intrusion detection, which is the first step, can be achieved automatically with machine learning methods. While Pham et al. [6] used the most important traditional machine learning methods such as decision tree, logistic regression, and random forest to detect an anomaly, the best result is achieved with logistic regression. The CSIC 2010 dataset, which is used in the evaluation of many studies and consists of requests coming to an e-commerce site, is used. In general, in this study, different studies are included in the step of feature extraction of the text.

In addition to such automatic anomaly detection studies, signature-based studies still produce effective results. Adem et al. [7] present a hybrid study using both signature-based and anomaly detection approaches. With a rule-based study based on character, word, and request length, a faster detection is made on CSIC 2010, ECML-PKDD 2007, and WUGD 2015 datasets with 95% accuracy.

Duy et al. [9], as a different approach, also consider user behavior in anomaly detection. In this study, the attributes of user behaviors on HTTP requests are determined. For feature extraction, TF-IDF and common feature approaches are used together with the random forest algorithm, and TF-IDF achieves a better result with a 4% difference.

Current malware acts as a benign request to evade attack detection. In this study [16], the difference between benign and malicious software is revealed by using a language-based approach to detect malicious requests. An importance score calculation is made between the characters of good and bad-tempered requests, and word order is created according to the order of importance. Then, a vector representation was created with doc2vec, which is a document-based approach to creating a feature vector.

Jemal et al. [18] made CNN, one of the best working deep learning algorithms, a three-step evaluation for each classification step, such as variations of the input vector and hyperparameter tuning. As the first evaluation step, the representation of the HTTP request as presentation and splitting is being worked on. As a second evaluation, different CNN hyperparameter combinations are compared, and as a final evaluation, the selection of the best deep learning toolbox. At the end of the study, the effect of the input vector and hyperparameter tuning on the CNN model is analyzed.

The biggest reason why detecting cyber attacks is problematic is that attacks are constantly being updated and become more complex. In order to cope with this problem, it is necessary to develop systems that are more flexible and adaptable to new types of attacks. Zhao et al. [17] have attempted to overcome this challenge with the semi-supervised Discriminant Autoencoder (AUE) classification method. The purpose of the algorithm is to ensure that unidentified samples get a place based on their nearest neighbors. In this way, it is to produce a generalized model by understanding the nature of the previously undefined attack.

Eduardo et al. [19] have also done a study as a solution to intrusion detection models that become obsolete and outdated over time. They provide anomaly detection with semi-supervised learning in order to update the existing model without human intervention. The experiments carried out within the scope of the study were carried out by considering the real network traffic of one year, and it is observed that the accuracy of the new data continues to be preserved. In the case of no model update, approximately 10% of the data is rejected, while in the case of a model update, this value becomes approximately 3%.

Bhati et al. [26] proposed an Intrusion Detection System (IDS) on an ensemble method using Extreme Gradient

Boosting (XGBoost), which provides the balance of bias-variance. The KDD-Cup99 dataset was used in the proposed model training and the obtained accuracy of the method is 99.95

Liu et al. [27] used the LightGBM ensemble technique to create a Network Intrusion Detection System (NIDS). By creating synthetic samples with the Adaptive Synthetic (ADASYN) oversampling method, the authors addressed the problem of class imbalance. In the suggested study, categorization is accomplished utilizing the datasets CICIDS2017, NSL-KDD and UNSW-NB15. According to their findings, LightGBM and ADASYN perform better than other traditional ML algorithms.

III. Method

Detecting malicious requests with high accuracy and high speed is a vital process in order to prevent systems from important damage. In our work, we aimed to find the best fit for this purpose. The techniques we applied in the experiments are divided into two parts. The first technique focuses on the classification of malicious and normal HTTP requests trained and tested by different models such as CNN, LSTM, XGBoost, LightGBM, and Random Forest. This study also focuses on fast detection of attacks as well as good accuracy results. The second technique focuses on new data upsampling algorithm for increasing the variety of normal requests since it aims to decrease disadvantages, for instance, lack of diversity and imbalance between classes.

A. Data

In this study, two different publicly available datasets have been taking considered. The first one is the CSIC dataset published and produced by the Spanish Research National Council (CSIC) [24], and it contains over 74000 requests with three different labels, which are normal, anomalous, Structured Query Language (SQL) injection, Cross-site scripting (XSS), Server-Side Includes Injection (SSI), buffer overflow, Carriage Return Line Feed injection (CRLFi), XML Path Language (XPATH) injection, Lightweight Directory Access Protocol injection (LDAPi), and format string. Dataset has been generated artificially via semi-automatic tools. Second dataset is the Discovery Challange Data [25] which was provided LIRMM (Laboratoire dInformatique, de Robotique et de Microelectronique de Montpellier, FRANCE) and the LGI2P (Ecole des Mines dAlcs, FRANCE) to use in ECML/PKDD Discovery Challenge. The dataset was named in this work as ECML. It was based on a dataset of real-world web traffic in conjunction with Bee Ware. It has eight labels which are normal, SQL injection, path traversal, command execution, LDAPi, XPATH injection, SSI injection and cross-site scripting. There are over 50000 requests in the dataset.

1) Data Preprocessing: Some important data preprocessing steps were applied to datasets in order to standardize the data since two different datasets were used to perform binary classification. Firstly, common header names were chosen,

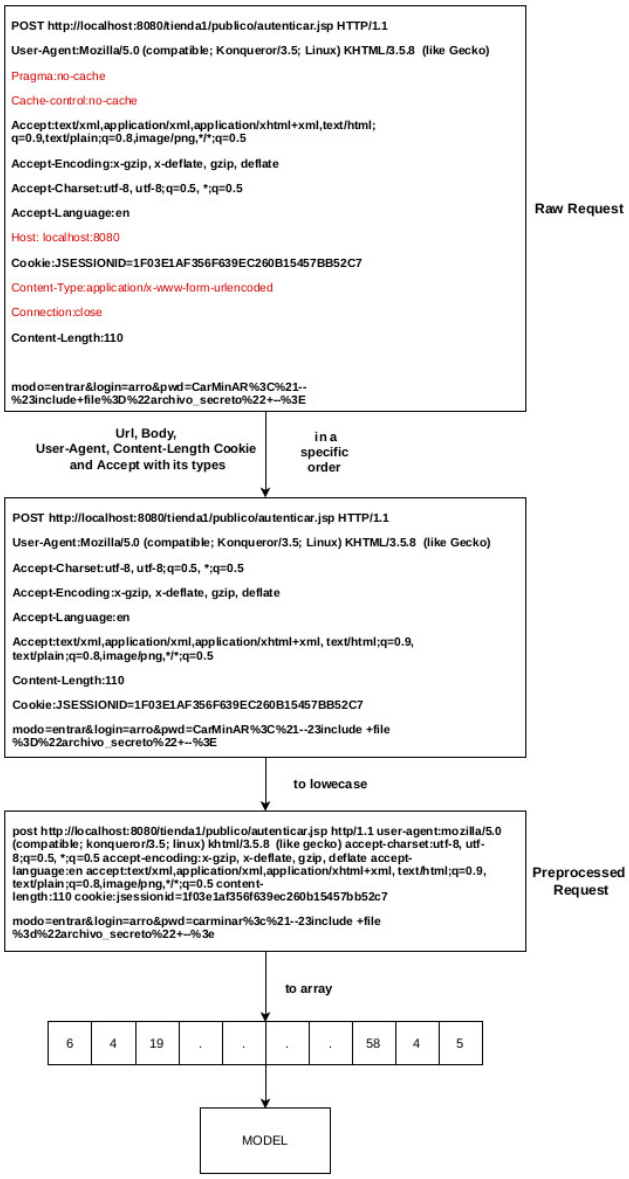

Fig. 1: An example of data preprocessing.

considering which header includes the attack in a specific order. More precisely, "User-Agent", "Accept", "Accept-Charset", "Accept-Encoding", "Accept-Language", "Content-Length", "Content-Encoding", and "Cookie" were used with the request type, URL, and the body part of the HTTP request. If there is no header in the original request, the header name will be added only the header name. Second, labels were changed to binary labels, i.e., normal requests labeled as "0" and different types of anomalous and malicious attack types labeled as "1". Finally, all characters turned into lowercase. See Fig. 1.

2) Data Upsampling: Data upsampling techniques can be used where a dataset has unbalanced data. In the appended dataset, we have 62.54% attack requests and 37.46% normal

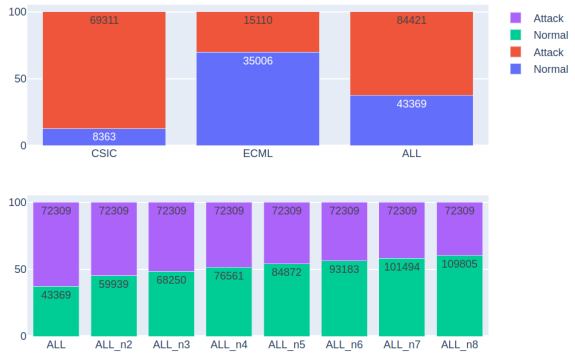

Fig. 2: Data distributions. In the red/blue part original dataset class distributions can be observed. In the purple/green part upsampled data class distributions can be observed. n_i where $i = 2, ..., 8$ means that CSIC's normal requests upsampled i times.

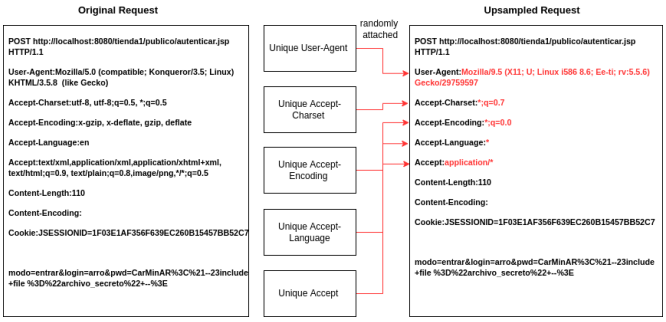

Fig. 3: An example of our upsampling method. While upsampling algorithm works, header values are randomly replaced from a unique list obtained from ECML for each header name. For example, "Accept-Language" header value was changed from "en" to "*".

requests, which refers to unbalanced data (see Fig. 2). We developed a new data upsampling technique for normal HTTP requests in order to provide a balanced dataset. Firstly, we found the number of unique header values for each header name considering labels (see Table I). As it can be seen in Table I, ECML has an excessive amount of unique values in both attack and normal requests. On the contrary, CSIC has a narrow range of unique values.

TABLE I: The numbers of unique header values

Header Name	Dataset Name	Attack	Normal
ACCEPT	ECML	4440	9044
	CSIC	6	2
ACCEPT Charset	ECML	5559	11738
	CSIC	2	1
ACCEPT Encoding	ECML	3257	6154
	CSIC	3	1
ACCEPT Language	ECML	6543	14210
	CSIC	3	1
User Agent	ECML	15075	35006
	CSIC	8664	1

Due to the few numbers of normal requests of CSIC and lacking unique values, we decided to upsample normal requests of this dataset using ECML's normal requests header values. The algorithm works in a way that randomly replaces the header values for each normal HTTP request of CSIC. See an example of replacing headers in Fig. 3. To examine if our upsampling technique works well and due to the number of samples we need, we applied this process for many times, i.e., 2 to 8 times. According to each dataset, normal and attack request percentages and the number of each label and total data of each upsampled dataset and the original datasets can be seen in Fig. 2. Moreover, each request converted character level arrays, i.e., each character has an integer value.

B. Models

Ensemble and deep learning methods are highly used in classification problems. In this paper, we would like to compare these methods' outcomes. Ensembles like RF, XGBoost, and LightGBM use a technique that combines different base models to produce one ideal predictive model. On the other hand, deep learning methods like CNN and LSTM provide automatic feature extraction techniques through filters/kernels and patterns across time, respectively, with learning parameters via forward and backward propagations. In our work, we applied input as character level which means that each character is translated into its corresponding integer. We divided the dataset into train, validation and test subsets with percentages of 60, 20, 20, respectively.

RF, XGBoost, and LightGBM classification models were fitted using default hyperparameters where estimators were evaluated using K-Fold cross-validation, where K is 5. Moreover, CNN and LSTM models are applied. Using hyperparameter tuning, decided hyperparameters are as follows. For the CNN model, the learning rate is 0.0001, the optimizer is Stochastic Gradient Descent (SGD), and filter and kernel sizes are 64 and 7, respectively. For the LSTM model, the learning rate is 0.001, and the optimizer is Adam. In Fig.4, each models layers can be seen. Additionally, both models are trained with batch sizes 32, and 5 epochs with an input size of 750.

IV. Results

Our work has two different outcomes. The first one is model-based and the second one is data-based outcomes. In model-based evaluation results, the binary LSTM classification model gave the best results in terms of accuracy, f1 score, precision, recall, and false-negative rate. Moreover, although LSTM is better than CNN except for the false-positive rate, CNN gave the best result on the evaluation of the false-positive rate. On the other hand, the RF classification model gave the worst results in terms of all numeric results, see Table II. Moreover, it can be easily seen that ensemble methods like XGBoost and LightGBM are as good as learning methods like CNN. However, RF was not able to perform like other methods even though it is a kind of ensemble algorithm. In

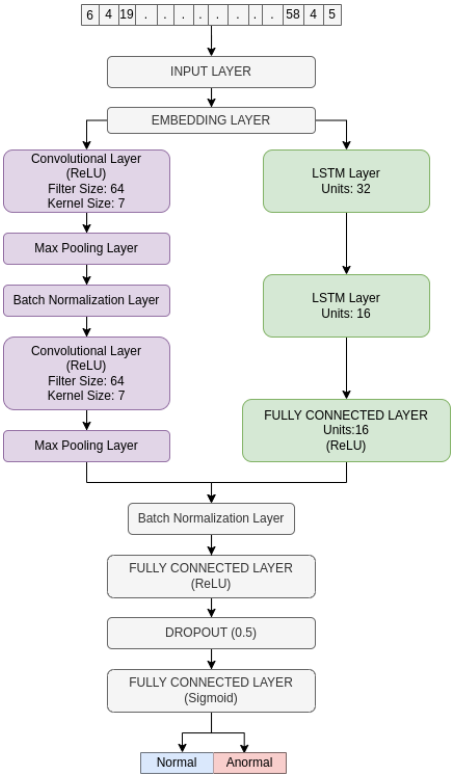

Fig. 4: Illustrations of CNN and LSTM classifier models. CNN and LSTM models have different layers. Separated by purple part shows CNN layers and separated by yellow part shows LSTM layers.

Table II, numeric results which is written in blue refers to the best results according to comparison type. Solely, the red refers to the worst results according to comparison type.

Secondly, we compared results between models and upsampled data (see Table II). In general, the LSTM model concluded that the best results are in accuracy, f1 score, precision, recall and false negative rate. However, some results of applications on false-positive rates CNN gave the best results. As expected from the model-based approach, the RF model gave the worst results for all numeric outputs. When we examine the training time outputs, we can easily observe that there is a vital difference between LSTM and LightGBM models (see Table II).

V. Conclusion

In this work, we tried to illustrate the effectiveness of different models which are CNN, LSTM, XGBoost, LightGBM and RF and to illustrate our new method for data upsampling and prove its efficiency. According to the results, LSTM is the best algorithm for character-level binary classification problems with HTTP requests. Additionally, ensemble methods such as XGBoost and LightGBM work as efficiently as CNN and LSTM. However, there is a serious trade-off between training time and the best model since LSTM average training time is approximately 85 times higher than LightGBM average training time. Furthermore, we explained our new data upsampling method for balancing datasets that have an unbalanced amount of classes. According to numerical results, our algorithm worked fine and proved its efficiency. In most cases, it improves the accuracy, f1 score, precision, recall, false-negative rate, and false-positive rate results.

VI. Future Work

Classification of HTTP requests as attack or not is essential for cyber security applications since hackers are waiting to steal any confidential information to use them to their advantage. Balancing the attack types in the dataset will increase the capability of detecting attacks for models. As a future work, we are planning to work on synthetic data generation and develop different upsampling methods for balancing attack types in the dataset.

References

[1] Isiker, B., and Sogukpinar, I. (2021). Machine learning based web application firewall. 2021 2nd International Informatics and Software Engineering Conference (IISEC). https://doi.org/10.1109/iisec54230.2021.9672335

[2] Pham, T. S., Hoang, T. H., and Van Canh, V. (2016). Machine learning techniques for web intrusion detection — a comparison. 2016 Eighth International Conference on Knowledge and Systems Engineering (KSE). https://doi.org/10.1109/kse.2016.7758069

[3] Li, J., Zhang, H., and Wei, Z. (2020). The weighted word2vec paragraph vectors for anomaly detection over HTTP traffic. IEEE Access, 8, 141787–141798. https://doi.org/10.1109/access.2020.3013849

[4] Ito, M., and Iyatomi, H. (2018). Web application firewall using character-level convolutional neural network. 2018 IEEE 14th International Colloquium on Signal Processing amp; Its Applications (CSPA). https://doi.org/10.1109/cspa.2018.8368694

[5] Bassett, G., Hylender, C. D., Langlois, P., Pinto, A., Widup, S. (2020). Verizon data breach investigations report. doi:10.13140/RG.2.2.21300.48008

[6] T. S. Pham, T. H. Hoang and V. Van Canh, "Machine learning techniques for web intrusion detection — A comparison," 2016 Eighth International Conference on Knowledge and Systems Engineering (KSE), 2016, pp. 291-297, doi: 10.1109/KSE.2016.7758069.

[7] Tekerek, A., Gemci, C., and Bay, Ö. F. (2016). Web tabanlı saldırı önleme sistemi tasarımı ve gerçekleştirilmesi: yeni bir hibrit model. Gazi Üniversitesi Mühendislik-Mimarlık Fakültesi Dergisi, 31(3). https://doi.org/10.17341/gummfd.63355

[8] Hoang, X. D. (2020). Detecting common web attacks based on machine learning using web log. Advances in Engineering Research and Application, 311–318. https://doi.org/10.1007/978-3-030-64719-3_35

[9] Duy, P. H., Thuy, N. T. T., and Diep, N. N. (2020). Anomaly detection system of web access using user behavior features. Southeast Asian J. Sciences, 7(2).

[10] Li, M., Wang, H., Yang, L., Liang, Y., Shang, Z., and Wan, H. (2020). Fast hybrid dimensionality reduction method for classification based on feature selection and grouped feature extraction. Expert Systems with Applications, 150, 113277. https://doi.org/10.1016/j.eswa.2020.113277

[11] Torrano-Gimenez, C., Nguyen, H. T., Alvarez, G., Petrovic, S., and Franke, K. (2011). Applying feature selection to payload-based web application firewalls. 2011 Third International Workshop on Security and Communication Networks (IWSCN). https://doi.org/10.1109/iwscn.2011.6827720

[12] Nguyen, H. T., Torrano-Gimenez, C., Alvarez, G., Petrović, S., and Franke, K. (2011). Application of the generic feature selection measure in detection of web attacks. Computational Intelligence in Security for Information Systems, 25–32. https://doi.org/10.1007/978-3-642-21323-6_4

[13] Li, H., Guo, W., Wu, G., and Li, Y. (2018). A RF-PSO based hybrid feature selection model in Intrusion detection system. 2018 IEEE Third International Conference on Data Science in Cyberspace (DSC). https://doi.org/10.1109/dsc.2018.00128

TABLE II: Model result with all datasets

Model Name	Data Name	Accuracy	F1 Score	Precision	Recall	FPRate	FNRate	Training Time(sec)
CNN	ALL	0.9549	0.9553	0.9595	0.9549	0.0014	0.0713	666.55
	ALL_n2	0.9666	0.9666	0.9684	0.9666	0.0040	0.0575	752.43
	ALL_n3	0.9678	0.9678	0.9695	0.9678	0.0028	0.0596	676.87
	ALL_n4	0.9710	0.9710	0.9722	0.9710	0.0031	0.0565	722.64
	ALL_n5	0.9692	0.9691	0.9703	0.9692	0.0044	0.0616	813.08
	ALL_n6	0.9756	0.9755	0.9761	0.9756	0.0058	0.0484	681.81
	ALL_n7	0.9777	0.9776	0.9781	0.9777	0.0053	0.0463	706.86
	ALL_n8	0.9780	0.9779	0.9784	0.9780	0.0039	0.0494	834.70
LSTM	ALL	0.9815	0.9816	0.9820	0.9815	0.0086	0.0243	3544.32
	ALL_n2	0.9867	0.9867	0.9869	0.9867	0.0035	0.0214	3987.73
	ALL_n3	0.9880	0.9881	0.9882	0.9880	0.0034	0.0199	4217.64
	ALL_n4	0.9886	0.9886	0.9887	0.9886	0.0043	0.0189	4489.10
	ALL_n5	0.9880	0.9880	0.9881	0.9880	0.0058	0.0192	4519.91
	ALL_n6	0.9860	0.9860	0.9860	0.9860	0.0140	0.0140	4750.85
	ALL_n7	0.9918	0.9917	0.9918	0.9918	0.0020	0.0171	5150.77
	ALL_n8	0.9927	0.9927	0.9927	0.9927	0.0030	0.0138	5617.30
XGBoost	ALL	0.9529	0.9532	0.9556	0.9529	0.0207	0.0627	47.16
	ALL_n2	0.9576	0.9577	0.9593	0.9576	0.0161	0.0639	56.82
	ALL_n3	0.9601	0.9601	0.9613	0.9601	0.0156	0.0628	62.98
	ALL_n4	0.9630	0.9629	0.9639	0.9630	0.0143	0.0613	67.61
	ALL_n5	0.9632	0.9631	0.9641	0.9632	0.0121	0.0657	55.49
	ALL_n6	0.9667	0.9666	0.9675	0.9667	0.0100	0.0633	63.81
	ALL_n7	0.9695	0.9694	0.9700	0.9695	0.0095	0.0600	82.74
	ALL_n8	0.9703	0.9702	0.9708	0.9703	0.0090	0.0612	106.40
LightGBM	ALL	0.9383	0.9389	0.9434	0.9383	0.0227	0.0847	51.51
	ALL_n2	0.9463	0.9464	0.9493	0.9463	0.0170	0.0839	40.59
	ALL_n3	0.9483	0.9483	0.9508	0.9483	0.0158	0.0855	53.76
	ALL_n4	0.9527	0.9526	0.9544	0.9527	0.0159	0.0808	55.59
	ALL_n5	0.9524	0.9522	0.9541	0.9524	0.0143	0.0868	58.44
	ALL_n6	0.9563	0.9561	0.9576	0.9563	0.0130	0.0833	54.13
	ALL_n7	0.9597	0.9595	0.9608	0.9597	0.0109	0.0815	57.15
	ALL_n8	0.9602	0.9599	0.9611	0.9602	0.0110	0.0836	57.23
RF	ALL	0.8986	0.8999	0.9099	0.8986	0.0437	0.1355	115.76
	ALL_n2	0.9149	0.9151	0.9220	0.9149	0.0271	0.1328	124.59
	ALL_n3	0.9179	0.9177	0.9238	0.9179	0.0252	0.1358	121.28
	ALL_n4	0.9243	0.9240	0.9285	0.9243	0.0241	0.1308	126.20
	ALL_n5	0.9265	0.9260	0.9306	0.9265	0.0201	0.1362	132.78
	ALL_n6	0.9304	0.9298	0.9338	0.9304	0.0184	0.1357	137.53
	ALL_n7	0.9350	0.9343	0.9376	0.9350	0.0177	0.1315	145.36
	ALL_n8	0.9372	0.9365	0.9393	0.9372	0.0181	0.1306	124.78

[14] J. Gupta and J. Singh, "Detecting anomaly based network intrusion using feature extraction and classification techniques," Int. J. Adv. Res. Comput. Sci., vol. 8, no. 5, pp. 1453–1456, 2017

[15] Liu, C., Yang, J., and Wu, J. (2020). Web intrusion detection system combined with feature analysis and SVM Optimization. EURASIP Journal on Wireless Communications and Networking, 2020(1). https://doi.org/10.1186/s13638-019-1591-1

[16] Mimura, M. (2020). Adjusting lexical features of actual proxy logs for intrusion detection. Journal of Information Security and Applications, 50, 102408. https://doi.org/10.1016/j.jisa.2019.102408

[17] Zhao, F., Zhang, H., Peng, J., Zhuang, X., and Na, S.-G. (2020). A semi-self-taught network intrusion detection system. Neural Computing and Applications, 32(23), 17169–17179. https://doi.org/10.1007/s00521-020-04914-7

[18] Jemal, I., Haddar, M. A., Cheikhrouhou, O., and Mahfoudhi, A. (2021). Performance evaluation of Convolutional Neural Network for web security. Computer Communications, 175, 58–67. https://doi.org/10.1016/j.comcom.2021.04.029

[19] Viegas, E. K., Santin, A. O., Cogo, V. V., and Abreu, V. (2020). A reliable semi-supervised Intrusion Detection Model: One year of network traffic anomalies. ICC 2020 - 2020 IEEE International Conference on Communications (ICC). https://doi.org/10.1109/icc40277.2020.9148916

[20] H. Zhang, X. Yu, P. Ren, C. Luo and G. Min, "Deep adversarial learning in intrusion detection: A data augmentation enhanced framework", arXiv:1901.07949, 2019

[21] Yuan, D., Ota, K., Dong, M., Zhu, X., Wu, T., Zhang, L., and Ma, J. (2020). Intrusion detection for smart home security based on data augmentation with Edge Computing. ICC 2020 - 2020 IEEE International Conference on Communications (ICC). https://doi.org/10.1109/icc40277.2020.9148632

[22] Wang, Y., lv, S., Liu, J., Chang, X., and Wang, J. (2020). On the combination of data augmentation method and gated convolution model for building effective and robust intrusion detection. Cybersecurity, 3(1). https://doi.org/10.1186/s42400-020-00063-5

[23] Farea, A. A., Wang, C., Farea, E., and Ba Alawi, A. (2021). Cross-site scripting (XSS) and SQL injection attacks multi-classification using bidirectional LSTM recurrent neural network. 2021 IEEE International Conference on Progress in Informatics and Computing (PIC). https://doi.org/10.1109/pic53636.2021.9687064

[24] Torpeda. (n.d.). Retrieved May 6, 2022, from https://www.tic.itefi.csic.es/torpeda/datasets.html

[25] Analyzing web traffic ECML/PKDD 2007 discovery challenge September 17-21, 2007, Warsaw, Poland. Attack Challenge - ECML/PKDD Workshop. (n.d.). Retrieved May 6, 2022, from https://www.lirmm.fr/pkdd2007-challenge/

[26] Bhati BS., Chugh G., Al-Turjman F., and Bhati NS. An improved ensemble based intrusion detection technique using XGBoost.Trans EmergTelecommun Technol. 2020.https://doi.org/10.1002/ett.4076

[27] Liu J., Gao Y., and Hu F. A fast network intrusion detection system using adaptive synthetic oversampling and LightGBM. Comput. Security 2021;106. doi:10.1016/j.cose.2021.102289