Using Transformer Models to Identify Security and Machine Learning Discussions of IoT Developers : A Comparative Study

Md. Rafi, Jerin Ahasan Kheya, Asif Mamun Hridoy, Syeda Annan Asrafi
Department of Computer Science & Engineering
Ahsanullah University of Science and Technology, Dhaka, Bangladesh
Email: {190104041,190104043,190104047,190104050}@aust.edu

***Abstract*—IoT devices are increasing everyday along with IoT developers. More developers are showing interest in IoT. Therefore, discussion about various IoT topics like 'security' and 'machine learning' are increasing in developer's discussion platform like StackOverflow. Automatic classification of these discussion can provide helpful insights to IoT vendors and they can learn about issues and current trends developers having while developing their IoT projects. Our study compares the performance of transformer models and showed an improved performance compared to previous works in classifying IoT security and machine learning related discussions. Our study found BERT performed better than RoBERTa in identifying both security and machine learning discussions.**

***Index Terms*—IoT, StackOverflow, machine-learning**

I. Introduction

Internet of Things (IoT) is an open and comprehensive network of intelligent objects that have the capacity to auto-organize, share information, data and resources, reacting and acting in face of situations and changes in the environment [1]. The number of IoT devices is increasing everyday. As of 2022, it is estimated that there were around 13.14 billion IoT devices in the world and this figure is expected to almost double to 29.42 billion by 2030 [2].

As IoT devices are everywhere around us (e.g., smart home devices, IP cameras, vehicle tracking devices etc.), vulnerabilities to the security of these devices can cause catastrophic damage to privacy of public. So, IoT developers are concerned about the security issues of IoT devices and they often discuss about these issues in various developers' discussion forums like Stack Overflow (SO) [3]. Moreover, the high volume of data produced by the IoT devices have created opportunities to adopt Machine Learning (ML) into IoT-based solutions [4]. Therefore, IoT developers are interested to adopt deep neural network-based ML models into their IoT devices, but they find it challenging to accommodate those into their resource-constrained IoT devices and they often seek help to online discussion forums like SO [5].

It is important to understand the challenges IoT developers face related to security and ML issues in order to design an effective techniques to address the challenges and also IoT vendors can support the developers according the problems they face [5]. Several works have been done for this purpose [3], [5]–[7]. Mandal et al. [3] created an IoT security dataset by manually labelling 7147 sentences from different IoT related SO posts and showed the effectiveness of using different transformer models to identify IoT security sentences. Uddin [5] created a dataset consisting both IoT related security and ML sentences from SO and used transformer model to classify security sentences but classified ML sentences by searching some specific keywords. Machine learning techniques could improve the performance of identifying ML sentences. We are aware of no such research that used machine learning techniques to classify both IoT related security and ML discussion from SO. The purpose of this study is to classify both IoT related security and ML sentences from SO posts using different transformer models and compare the performance of these models.

II. Related Work

Many studies have been conducted to analyze SO discussions. Barura et al. [8] used topic modeling to analysis of topics and trends in SO. Uddin et al. [6] applied topic modeling to find what IoT topics are discussed by developers on SO along with other research questions. These studies shows that popularity of IoT discussion in SO is increasing.

Some studies have been conducted to detect security and ML discussion of IoT developers from SO. Mandal et al. [7] developed a model (SecBot) that can automatically find security related IoT discussions in SO to understand the challenges IoT developers face while applying security practices and techniques to IoT devices. SecBot outperformed all other models in the study and gave F1-Score of 0.935.

Mandal et al. [3] used different transformer models (BERT [9], RoBERTa [10], XLNET [11] and BERTOverflow [12]) to detect IoT security sentences from SO posts. The study found that RoBERTa model has the best F1-Score of 0.69 for the purpose.

Uddin [5] studied how do developers discuss security and ML issues in SO. The study found that RoBERTa was the best performing model to identify IoT security related sentences with an F1-score of 0.91. For ML IoT ML sentences detection, the study relied on a set of specific keywords.

III. Dataset Description

Mandal et al. [3] created a dataset consisting IoT security sentences from SO posts. The dataset uses the SO data dump of September 2021. The dataset is manually labeled by three developers separately and resolve the conflict by majority voting. The dataset contains a total of 7147 sentences. Among them 250 sentences are labeled as 'Security = 1' (contains security aspect).

Gias Uddin [5] created a dataset that contains both IoT related security and machine learning sentences from SO data dump upto September 2019. The dataset contains 672,678 sentences. Each sentence is labeled HasSecurity=1 if the sentence contains security related information and HasSecurity=0 otherwise. Similarly, a sentence is labeled HasML=1 if that sentence contains machine learning related discussion or HasML=0 otherwise. The dataset has 30,192 labeled as security related and 801 sentences labeled as ML related.

For this study the dataset [3] is used for the purpose of detecting IoT security sentences. For ML sentence detection, we create a dataset by taking 801 ML sentences and 2399 non ML sentences form the dataset [5].

The summery of the datasets used in this study is given in Table I.

TABLE I
SECURITY AND ML DATA DISTRIBUTION

Dataset	Size	Security	ML
For security [3]	5919	1049	-
For ML	3200	-	801

Both security and ML dataset is first split into 80-20 ratio. The 20% data used as test data. The other 80% data is further split into 80-20 ratio. This time the 80% split data is used for training and the 20% split data is used for validation. The distribution of datasets splits into train, test, and validation is given in Table II. In this study, we have used the same set of train, test and validation data for all models.

TABLE II
DATASET SPLIT DISTRIBUTION

		Dataset	
		Security	**ML**
Train	Label 0	3117	1535
	Label 1	671	513
	Total	**3788**	**2048**
Validation	Label 0	779	384
	Label 1	168	128
	Total	**947**	**512**
Test	Label 0	974	480
	Label 1	210	160
	Total	**1184**	**640**

IV. Methodology

The key steps proposed methodology is shown in Figure I. Short description of these steps are given here:

Input: Here, inputs are the documents from the corpus. The raw documents are considered as input. For demonstration purpose, let consider that the document "*If the LTPA token is not yet present the user must authenticate normally.*" from security training dataset is current input.

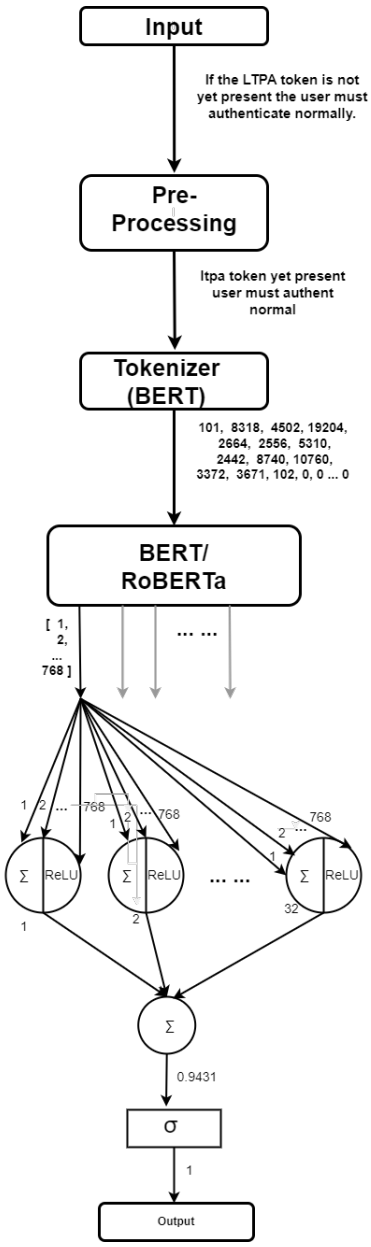

Fig. 1. Flow diagram of proposed methodology

Pre-processing: Each input sentences is go throgh pre-processing. Pre-processing includes making sentences lowercase, removing punctuations, stop words and numbers. Finally, stemming is applied to remove last last few characters from words. For example, the processed output from the previous input example will be- "*ltpa token yet present user must authent normal*".

Tokenization: After pre-processing, each document is tokenized using BERT tokenizer. The length of each documents is 150. The BERT tokenizer adds [CLS] and [SEP] tokens

at the start and end of of each documents respectively. For example, after tokenization, the previous processed input will be a vector consisting- "*[101, 8318, 4502, 19204, 2664, 2556, 5310, 2442, 8740, 10760, 3372, 3671, 102, 0, 0, 0 ... 0]*". The length of the vector will be 150. If the processed input has less than 150 words, then extra 0 will be added like this example and if the processed input has more than 150 words then all the words after length 150 will be truncated.

Transfomer Models: In this step the sequence of tokens is provided as the input for transformer models. We have used two transformer models so far- BERT and RoBERTa. The output of this step is a vector for each input tokens. The size of the vector is made with 768 float numbers.

Fully Connected Neural Network: From the previous output, only the first vector i.e. the output vector for [CLS] token is the input vector for the network. All other output vectors are left out. Hidden layer of the network is consists of 32 hidden nodes and output layer is consist of one output node. ReLU is used as the activation funcction in the hidden layer and sigmoid is used in the output layer to classify a sentence as 0 or 1. For example, the fist output vector from the BERT output is fed into the network and output layer return value 0.9431 for that sentence. This value is then passed to sigmoid actiavtion function, and we get an output 1. The ground truth of the sentence was also 1.

Hyper parameters used for both BERT and RoBERTa models in this study are: batch size = 8, learning rate = 1e-5, number of epochs = 5.

V. Result Analysis

The confusion matrix for the security test dataset for BERT and RoBERTa is given in Fig. 2 and Fig. 3 respectively.

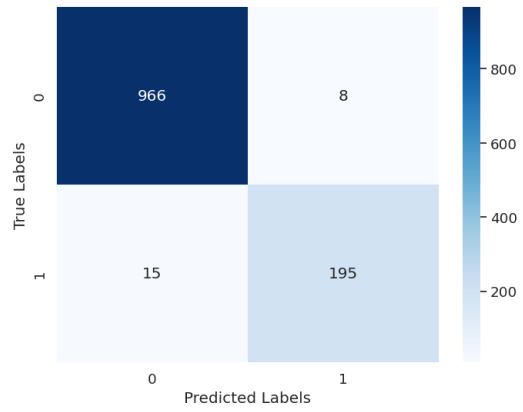

Fig. 2. Confusion Matrix for Security Dataset(BERT)

The confusion matrix for the ML test dataset for BERT and RoBERTa is given in Fig. 4 and Fig. 5 respectively.

The accuracy, precision, accuracy, F-1 score for both models and both datasets is given in Table III and Table IV respectively.

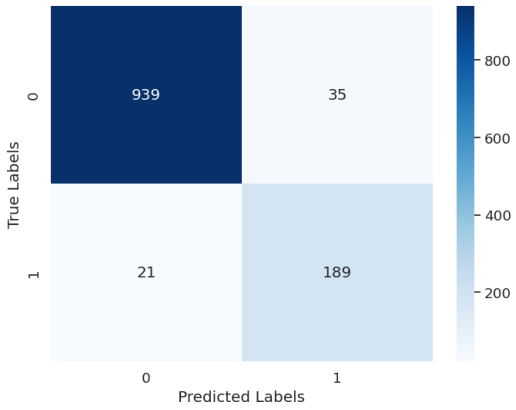

Fig. 3. Confusion Matrix for Security Dataset(RoBERTa)

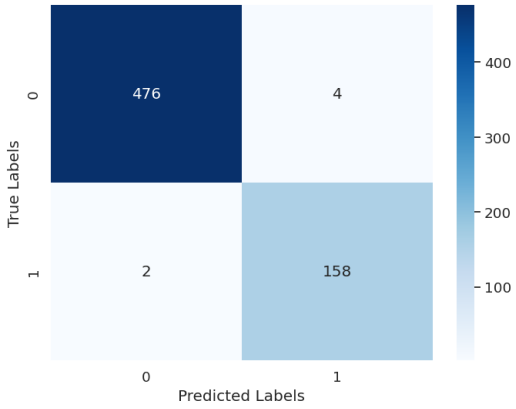

Fig. 4. Confusion Matrix for ML Dataset(BERT)

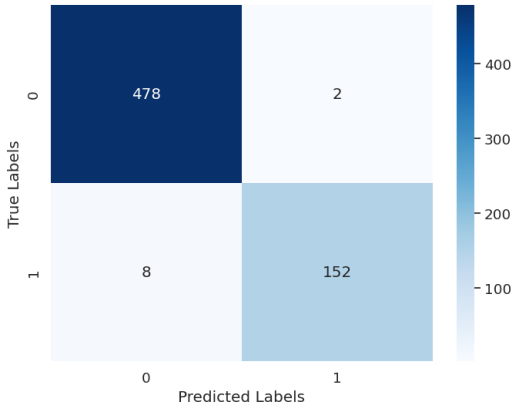

Fig. 5. Confusion Matrix for ML Dataset(RoBERTa)

TABLE III
Performance of models for Security Dataset

BERT	Accuracy	**0.980**
	Precision	**0.960**
	Recall	**0.928**
	F-1 Score	**0.943**
RoBERTa	Accuracy	0.952
	Precision	0.843
	Recall	0.90
	F-1 Score	0.870

TABLE IV
PERFORMANCE OF MODELS FOR ML DATASET

BERT	Accuracy	**0.990**
	Precision	0.975
	Recall	**0.987**
	F-1 Score	**0.980**
RoBERTa	Accuracy	0.984
	Precision	**0.987**
	Recall	0.950
	F-1 Score	0.968

As we can see, the overall score for the ML dataset is better than the security dataset for both models. Uddin's [5] dataset contains ML sentences that has one or more specific keywords in them. We took these sentences to create our ML dataset. So, all the ML sentences in our ML dataset has contains at least one of the keyword. Therefore, the differences between ML and non-ML sentences in the dataset is very clear. So after some training it was clear to the model that the sentences containing those keywords are ML sentences. Therefore, the performance in ML dataset for both models are better than the security dataset.

It also can be seen that BERT outperformed RoBERTa in identifying both security and ML sentences except precision for BERT in security dataset is lower than RoBERTa.

Uddin's [5] best performing classifier for security dataset [5] was RoBERTa which showed an F1-score of 0.91. Mandal et al. [3] fine tuned BERT and RoBERTa showed an F1-score of roughly .65 and .70 respectively for the security dataset. Our best proposed model for security dataset has F1-score of 0.94 which is an improvement.

VI. FUTURE WORKS

The ML dataset has few data and all the sentences in the dataset has some common keywords among them. But, in reality not all the ML sentences will contain these keywords. So, the ML dataset can be expanded by adding various ML sentences. The variance between the sentences should be larger in order to perform the model good in practical situation. The security dataset can be expanded and can be balanced by adding more security sentences. Other models can be analyzed and compare their performances. Comparison between more transformer models can study in future. Both deep learning models and traditional machine learning models can be used for the purpose of the study. Due to the time constraints we could not perform these tasks. These are the opportunities that can be perform in future to extend this study.

VII. CONCLUSION

This study has shown that transformer models has shown satisfactory performances for identifying IoT security and ML sentences from SO discussions. Automatic identification of IoT security and ML sentences will help IoT vendors to understand what problems IoT developers faced during development. IoT vendors then can support the developers. However, the performance of these models for a more diverse dataset is still unknown.

REFERENCES

[1] Madakam, S., Lake, V., Lake, V., & Lake, V. (2015). Internet of Things (IoT): A literature review. Journal of Computer and Communications, 3(05), 164.

[2] Vailshery, L. (2022). IoT connected devices worldwide 2019-2030. Statista. https://www.statista.com/statistics/1183457/iot-connected-devices-worldwide

[3] Mandal, N. C., Shahariar, G. M., & Shawon, M. T. R. (2023, January). Effectiveness of Transformer Models on IoT Security Detection in Stack-Overflow Discussions. In Proceedings of International Conference on Information and Communication Technology for Development: ICICTD 2022 (pp. 125-137). Singapore: Springer Nature Singapore.

[4] Marjani, M., Nasaruddin, F., Gani, A., Karim, A., Hashem, I. A. T., Siddiqa, A., & Yaqoob, I. (2017). Big IoT data analytics: architecture, opportunities, and open research challenges. ieee access, 5, 5247-5261.

[5] Uddin, G. (2021, June). Security and machine learning adoption in IoT: A preliminary study of IoT developer discussions. In 2021 IEEE/ACM 3rd International Workshop on Software Engineering Research and Practices for the IoT (SERP4IoT) (pp. 36-43). IEEE.

[6] Uddin, G., Sabir, F., Guéhéneuc, Y. G., Alam, O., & Khomh, F. (2021). An empirical study of iot topics in iot developer discussions on stack overflow. Empirical Software Engineering, 26, 1-45.

[7] Mandal, N., & Uddin, G. (2022). An empirical study of IoT security aspects at sentence-level in developer textual discussions. Information and Software Technology, 150, 106970.

[8] Barua, A., Thomas, S. W., & Hassan, A. E. (2014). What are developers talking about? an analysis of topics and trends in stack overflow. Empirical software engineering, 19, 619-654.

[9] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

[10] Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., ... & Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692.

[11] Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R. R., & Le, Q. V. (2019). Xlnet: Generalized autoregressive pretraining for language understanding. Advances in neural information processing systems, 32.

[12] Tabassum, J., Maddela, M., Xu, W., & Ritter, A. (2020). Code and named entity recognition in stackoverflow. arXiv preprint arXiv:2005.01634.