Comparison of XGBoost and RoBERTa for Botnet Network Traffic Detection in an IoT Context

Submitted by: Paul Larison

Faculty Advisor:
Dr. Berleant
Dr. Tudoreanu
Dr. Wu

Date Submitted:
5/8/23

Table of Contents

Abstract

This project compares two machine learning models for botnet traffic detection on IoT networks: XGBoost, a gradient-boosted decision tree, and RoBERTa, a transformer-based neural network. The comparison is carried out on the IoT-23 dataset, which contains network traffic captures from IoT devices and is intended to provide researchers with real, labeled IoT malware infections and benign traffic for developing machine learning algorithms. The models were trained on a training set of 10 samples and tested on 10 of the remaining samples. Both models had near-identical F1 scores on six of the ten test sets. XGBoost outperformed RoBERTa on three of the remaining four sets. XGBoost processing is also considerably faster than RoBERTa. The project concludes that RoBERTa is capable of this task and may be suited as a component in an ensemble detection system, but XGBoost is superior as a stand-alone model.

Introduction

The increasing popularity of the Internet of Things (IoT) has resulted in a massive surge in the number of connected devices. IoT devices are integrated into various aspects of our lives, including home automation, healthcare, transportation, and industrial control systems. This widespread adoption of IoT devices has also increased the risk of cyber threats, including botnets, which are networks of compromised devices controlled by an attacker to perform malicious activities. These activities include distributed denial of service attacks (DDoS), spamming, and identity theft. As a result, detecting botnet activity in IoT traffic has become a crucial task for ensuring the security of the IoT ecosystem.

Considerable research has been done on using machine learning to detect such activity, especially on methods that use traditional, non-deep learning methods. Boosted decision trees such as XGBoost are generally regarded as having the best performance for this purpose. Elsewhere in the world of AI, attention-based transformers have made significant contributions to natural language processing, as well as other fields, including computer vision and audio processing, but have not been heavily studied in a network security context.

This project focuses on the use of transformers in that context, specifically in detecting botnet activity in IoT traffic. It compares the performance of a traditional machine learning method, XGBoost, with that of RoBERTa, a state-of-the-art transformer-based model, to explore their respective strengths and limitations in this context. Their performance will be evaluated using accuracy, precision, recall, F1 scores, confusion matrices and ROC curves. Tuning, training and testing times will also be compared.

Objectives and Scope

The main objective of this project is to compare the performance of two machine learning models, XGBoost and RoBERTa, in detecting botnet activity in IoT traffic. The project aims to explore the following research questions:

- How well does RoBERTa perform in detecting botnet activity in IoT traffic compared to XGBoost?
- Is there any difference in performance when classifying bots that are included in the training set vs. bots that are unknown to the models?

- How do the models compare in terms of processing time?

To achieve these objectives, the project will use the IoT-23 dataset, which contains labeled traffic captures from IoT devices. The dataset will be split into a single training and multiple testing sets. Both models will be tuned and trained on the same training set, then tested and evaluated.

This project does not seek to determine whether RoBERTa is the optimum transformer model for this task, nor does it attempt to learn why the models perform better on some test sets than others.

Tools Used

- Anaconda [3] - Specifically the Spyder IDE for Python. This will run on my local desktop and be used for data pre-processing.
- Google Colab [4] - Web-based Python IDE. This will be used for transformer related tasks.
- HuggingFace [5] - Repository for NLP transformers and related datasets
- ChatGPT [6] - Code snippet generation.

Literature Review

The Internet of Things (IoT) has become an essential part of our daily lives, connecting numerous devices to the internet and facilitating communication among them. Inadequate security design has rendered these devices vulnerable to cybercriminals, resulting in the emergence of malware in IoT-connected devices. Vignau et al. [1] conducted a survey on the evolution of IoT malware in the past decade,

focusing on 16 of the most harmful IoT malware programs published between 2008 and 2018. They classified the malware based on their behavioral features, means of achieving their goals, and mechanisms to conceal their presence. Mohammadi et al. [2] discussed the potential of deep learning techniques for analytics and learning in the IoT domain. They explained why deep learning is a promising approach for achieving desired analytics in IoT big data and streaming data analytics and discussed the potential challenges associated with this approach.

Botnet attacks routinely exploit vulnerabilities in network protocols to perform malicious activities such as Distributed Denial of Service (DDoS) attacks and click fraud [4]. Network Intrusion Detection Systems (NIDS) are crucial for protecting IoT devices. The proposed NIDS approaches leverage machine learning algorithms to detect botnet attacks against protocols such as DNS, HTTP, and MQTT in IoT network traffic. Ensemble-based approaches using conventional machine learning currently provide the best results [3][5]. For instance, Moustafa et al. [3] proposed an ensemble-based NIDS that uses AdaBoost learning to evaluate the effectiveness of new statistical flow features extracted from network protocols. The proposed ensemble approach achieved higher detection rates and lower false-positive rates compared to individual techniques. Similarly, Sivasankari and Kamalakkannan [5] proposed an ensemble-based NIDS that combines Decision Tree, Support Vector Machine, and Naïve Bayes algorithms to enhance the detection rate and reduce the false-positive rate. The proposed NIDS techniques can be deployed in fog nodes to reduce the overhead of each node in the IoT network [5].

The Transformer architecture, based solely on attention mechanisms, has emerged as the dominant model in many fields, achieving state-of-the-art results while requiring less training time than previous deep learning techniques[7]. Model pretraining has allowed the Transformer architecture to be trained on generic corpora and then adapted to specific tasks with strong performance gains. The Hugging Face Transformers library was created to provide support for Transformer-based architectures and facilitates the distribution of pretrained models [8]. One such model is RoBERTa, an improved BERT pretraining approach that extends these performance gains by training the model longer with larger batches over more data and removing the next sentence prediction objective, among other modifications [9]. Transformers do not yet dominate every machine learning category; while deep learning models have been proposed for tabular data, including the TabTransformer model [11], studies have shown that decision tree models such as XGBoost outperform deep learning models while requiring less tuning [10].

The most similar work I could find to this project was done by Raghavendra and Chen [15], who compared the performance of decision trees, namely XGBoost with other machine learning methods, including Kitsune, an autoregressive deep learning-based method. They showed that decision trees are highly accurate (99.99%) and have less training and prediction time than Kitsune and most other machine learning methods. They used the same IoT-23 dataset, but did not test for unknown bots, as their training and test sets were constructed from just two types of bots and additional captures were not tested. Hegde et al. [16] also used the IoT-23 set to test algorithm performance. They mostly determined that large, diverse datasets work

better for this purpose. They also only tested against partitions of their training sets. Another similar project was done by Xu and Zheng [17], who created a hybrid model for network anomaly detection using gradient boosting decision trees and TabTransformer-based classification models.

Methodology

Dataset Selection and Preprocessing

The dataset used in this project is IoT-23, a collection of network traffic captures from Internet of Things (IoT) devices. It consists of 20 malware captures and 3 captures of benign IoT devices traffic, and was published in January 2020 by the Stratosphere Laboratory at CTU University, Czech Republic. The goal of the dataset is to provide researchers with real and labeled IoT malware infections and benign traffic for developing machine learning algorithms. The malicious captures were executed in a controlled network environment using a Raspberry Pi, while the benign captures were obtained from real IoT devices (a Philips HUE smart LED lamp, an Amazon Echo home intelligent personal assistant, and a Somfy smart doorlock). The dataset includes network data for each capture, as well as the protocols found in each capture. Additionally, the project used Zeek, a passive network traffic analyzer, to filter and summarize application layer protocol predictions for each infected device.

There are other datasets containing packet capture data that is processed with Zeek or a similar tool and then labeled. There is unfortunately no consistency in the generated features. IoT-23 was selected largely because it is a collection of separate captures with a consistent format.

Preprocessing was done on my local desktop to cut down on file transfer times. A script was created to extract each capture from its location in the .zip archive and convert it to a dataframe with a header and consistently formatted labels. Unused features, such as uid and ip addresses, were also removed during this step. The data was then cleaned and, with the exception of a few captures that were already balanced, ran through a function that undersampled the predominant label (usually 'malicious') in order to create more balanced sets to work with.

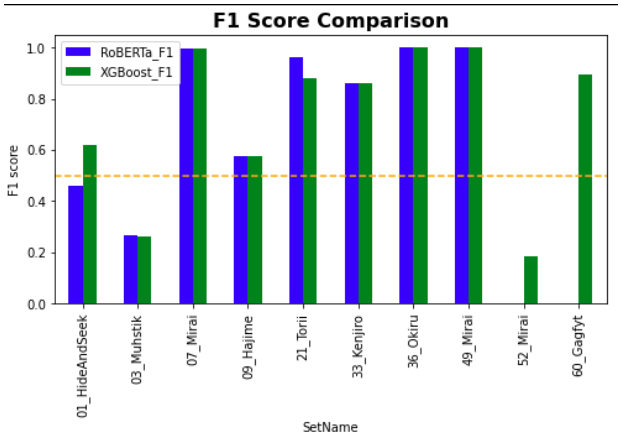
Once the resampled datasets were complete, two more functions were used to convert the data to the format expected by the two learning models being used. The XGBoost formatted data was uploaded to Google Drive in .csv form, while the RoBERTa formatted data was converted to Huggingface datasets before being uploaded and tokenized.

Model Training and Evaluation

The training and testing were done using Google Colab Pro+. The performance of the XGBoost and RoBERTa models were recorded and analyzed using the evaluation metrics mentioned above. Hyperparameter tuning was performed on both models. For XGBoost, the hyperparameters tuned were max_depth, min_child_weight, gamma, learning_rate, subsample, and colsample_bytree. RoBERTa has fewer hyperparameters available, but the tuning process takes more time. The RoBERTa hyperparameters tuned were batch_size, epochs, learning_rate and weight_decay.

After tuning, the models were trained using a training set that was created using 10 of the 23 samples. This included all three benign sets and the bots represented by multiple sets, specifically Mirai (x5), Kenjiro, and Torii. These are considered "known" bots. The remaining samples were used as test sets, which included "unknown" bots such as Hide and Seek, Muhstik, Hajime, Okiru, and Gagfyt. The Trojan and IRCBot samples were discarded due to their small sample size. The training data was split into a 70/30 ratio using random_state=7 for training and validation.

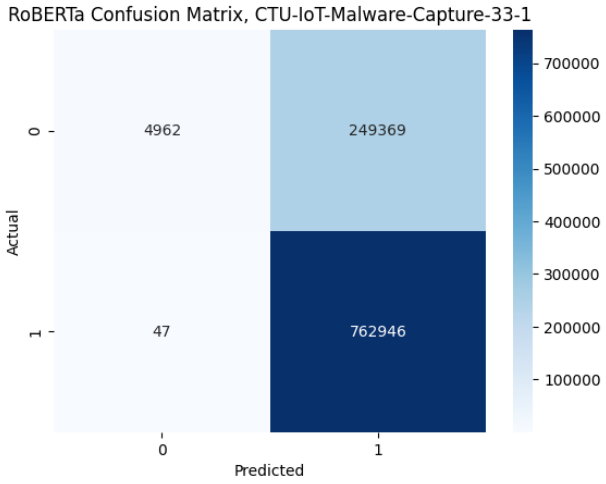
Results

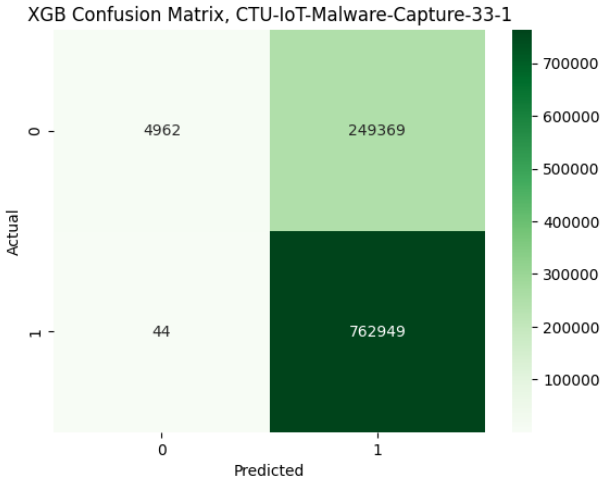


The study tested two models, XGBoost and RoBERTa, on several bot datasets, with the aim of identifying malicious network traffic. The models were evaluated based on their performance, primarily measured by the F1 score, on each dataset. The datasets included both known and unknown bots, and the models were expected to perform better on the known bots that were part of the training set.

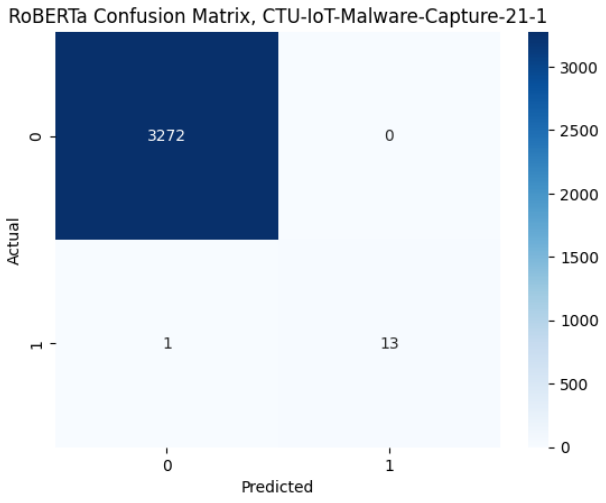
The known bots included Mirai, Kenjiro, and Torii, and the models performed well on two of the Mirai test sets, with F1 scores above 0.997. However, they performed poorly on the third Mirai set, with RoBERTa failing to classify a single malicious flow and XGBoost managing only a low F1 score of 0.18377. It is unclear why this Mirai dataset was different from the others, and this is beyond the scope of this project.

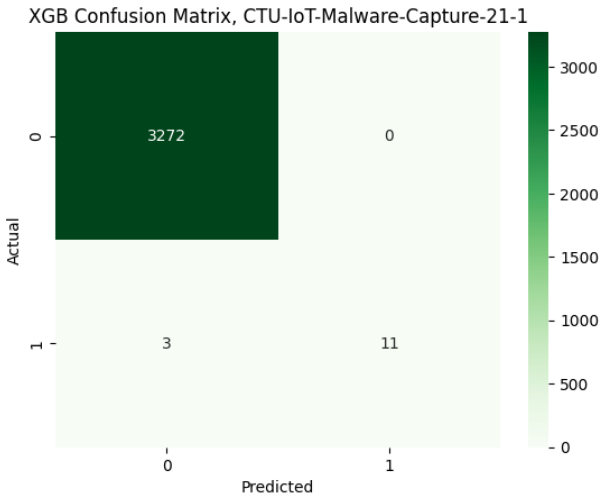
On the Kenjiro test set, both models had identical F1 scores of 0.85951, but a closer look showed that there were more false positives than correctly identified flows.





The Torii dataset had a small sample size of only 14 malicious flows, and was retained only because there are only 3 bots with multiple captures. RoBERTa was slightly better at detecting these, only misclassifying 1 flow.





The unknown bots included Okiru, Muhstik, Hajime, Hide and Seek, and Gagfyt. Okiru detection was a success for both models, with F1 scores above 0.99. Both models performed poorly on Muhstik, with F1 scores of 0.26710 and 0.26074 for RoBERTa and XGBoost, respectively. Hijime performance was mutually mediocre, with scores that both rounded to 0.575. On Hide and Seek, XGBoost performed better than RoBERTa, with an F1 score of 0.62 compared to 0.46. On Gagfyt, RoBERTa failed entirely, predicting only 5 benign flows (incorrectly), while XGBoost had an F1 score of 0.89.

Training time was also considered. On a Colab instance configured to use premium GPUs, RoBERTa took 103 minutes to tune and 12 minutes per epoch (36 minutes total) to train. Processing the test sets required an amount of time per line comparable to training. Fortunately the same model could be used in all tests. XGBoost required a new model for every variation in columns caused by one-hot encoding, but since it could tune and train in under a minute it was still much faster than RoBERTa.

In summary, both models achieved high F1 scores on some bots, notably the over-represented Mirai, but struggled with others. XGBoost performed better in most cases when there was a performance difference. XGBoost also had much shorter training times.

Conclusion

Based on the results of the study, it appears that XGBoost is better at identifying malicious traffic than RoBERTa, but both models struggled with some of the unknown

bot datasets. This suggests that a single model may not be enough to effectively detect all types of malicious traffic.

One possible next step could be to use an ensemble model to combine the strengths of both models to improve detection accuracy. The ROC plots also suggest that there may be additional variation between model predictions, further supporting the idea of using an ensemble approach. Additional models not used in this project may also be incorporated.

Another potential avenue for future research is to explore the reasons behind the performance differences observed between the two models on certain datasets. Investigating the characteristics of the data and the features that the models are using to make predictions could lead to insights on how to improve the models' performance. Additionally, further experimentation with hyperparameter tuning and feature selection could help to optimize the models for detecting specific types of malicious traffic.

Overall, while the study provides some promising results, it also highlights the challenges of accurately detecting all types of malicious network traffic. The development of effective models for detecting and preventing cyber attacks is an ongoing and important area of research, with implications for both individual and societal security.

References

1. Sebastian Garcia, Agustin Parmisano, & Maria Jose Erquiaga. (2020). IoT-23: A labeled dataset with malicious and benign IoT network traffic (Version 1.0.0) [Data set]. Zenodo. http://doi.org/10.5281/zenodo.4743746
2. *Anaconda Software Distribution*. Computer software. Vers. 2022.10. Anaconda, Nov. 2016. Web. https://www.anaconda.com.
3. Google Colaboratory (2017) Google colab. Google. Available at: https://colab.research.google.com/ (Accessed: January 23, 2023).
4. Wolf, Thomas, et al. "Huggingface's transformers: State-of-the-art natural language processing." *arXiv preprint arXiv:1910.03771* (2019).
5. OpenAI. “ChatGPT.” OpenAI, 2021, https://chat.openai.com
6. Vignau, Benjamin, Raphaël Khoury, and Sylvain Hallé. "10 years of IoT malware: A feature-based taxonomy." *2019 IEEE 19th International Conference on Software Quality, Reliability and Security Companion (QRS-C)*. IEEE, 2019.
7. Mohammadi, Mehdi, et al. "Deep learning for IoT big data and streaming analytics: A survey." *IEEE Communications Surveys & Tutorials* 20.4 (2018): 2923-2960.
8. Dhayal, Himanshi, and Jitender Kumar. "Botnet and p2p botnet detection strategies: a review." *2018 International Conference on Communication and Signal Processing (ICCSP)*. IEEE, 2018.
9. Moustafa, Nour, Benjamin Turnbull, and Kim-Kwang Raymond Choo. "An ensemble intrusion detection technique based on proposed statistical flow

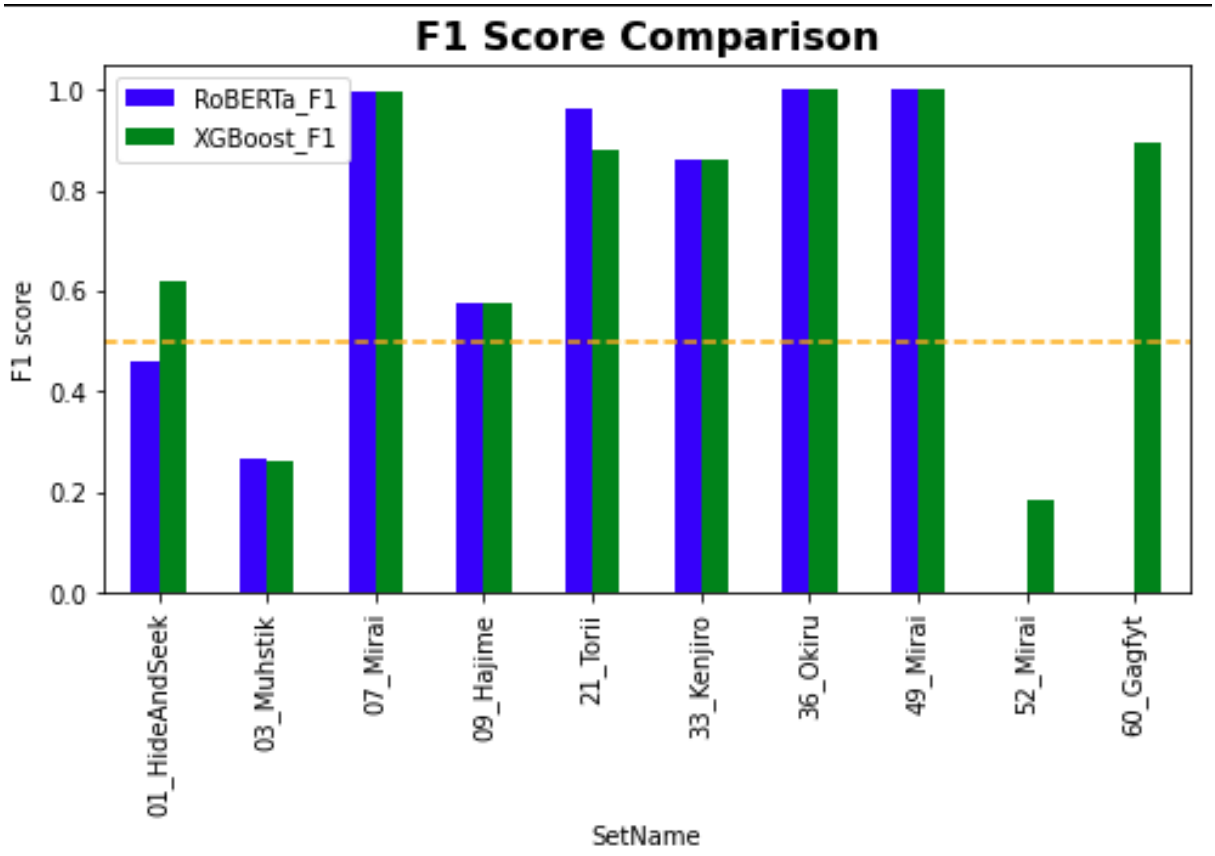
features for protecting network traffic of internet of things." *IEEE Internet of Things Journal* 6.3 (2018): 4815-4830.

10. Sivasankari, N., and S. Kamalakkannan. "Building NIDS for IoT Network using Ensemble Approach." *2022 7th International Conference on Communication and Electronics Systems (ICCES)*. IEEE, 2022.
11. Vaswani, Ashish, et al. "Attention is all you need." *Advances in neural information processing systems* 30 (2017).
12. Liu, Yinhan, et al. "RoBERTa: A robustly optimized BERT pretraining approach." *arXiv preprint arXiv:1907.11692* (2019).
13. Huang, Xin, et al. "Tabtransformer: Tabular data modeling using contextual embeddings." *arXiv preprint arXiv:2012.06678* (2020).
14. Shwartz-Ziv, Ravid, and Amitai Armon. "Tabular data: Deep learning is not all you need." *Information Fusion* 81 (2022): 84-90.
15. Raghavendra, Meghana, and Zesheng Chen. "Detecting IoT Botnets on IoT Edge Devices." *2022 IEEE International Conference on Communications Workshops (ICC Workshops)*. IEEE, 2022.
16. Hegde, Mandira, et al. "Identification of botnet activity in IoT network traffic using machine learning." *2020 International conference on intelligent data science technologies and applications (IDSTA)*. IEEE, 2020.
17. Xu, Xinyue, and Xiaolu Zheng. "Hybrid model for network anomaly detection with gradient boosting decision trees and tabtransformer." *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021.

Appendix A: Colab Folder Link

https://drive.google.com/drive/folders/1PtNdZrD_BnEROcVM3bBQGd6PQ4udc_yW?usp=sharing

Appendix B: Detailed Results

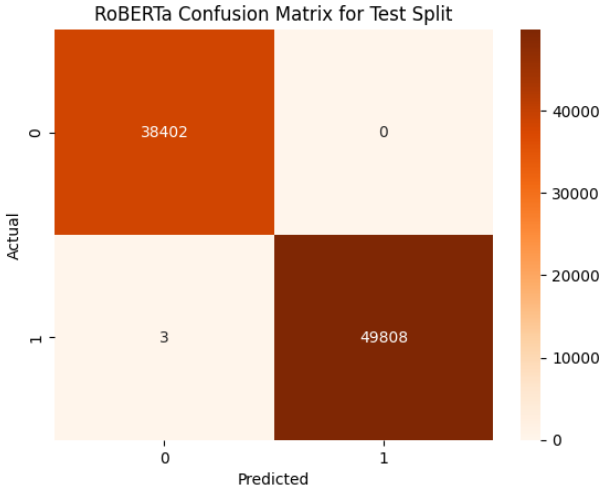


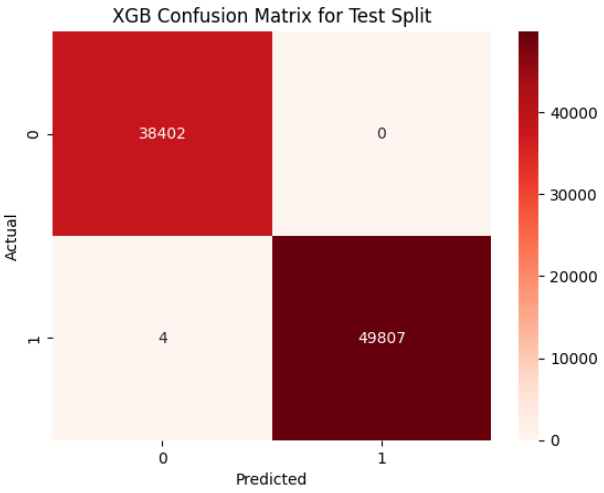
00_TestSplit

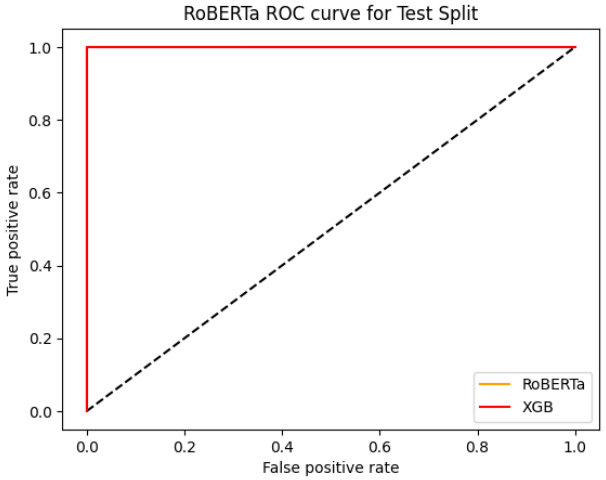
[Test Split]	RoBERTa	XGB
Accuracy	0.99997	0.99995
Precision	1.00000	1.00000
Recall	0.99994	0.99992
F1-score	0.99997	0.99996

Label Counts

[Test Split]	Actual	RoBERTa	XGB
1	49811	49808	49807
0	38402	38405	38406





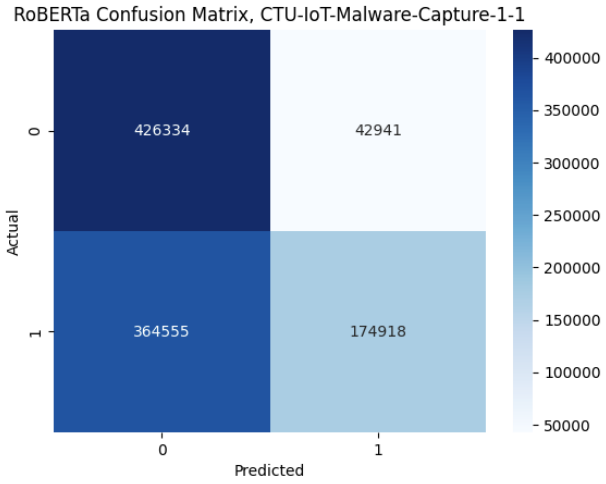


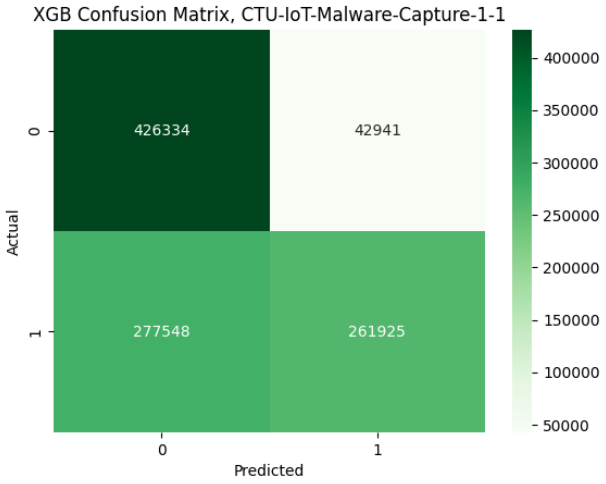
01_HideAndSeek

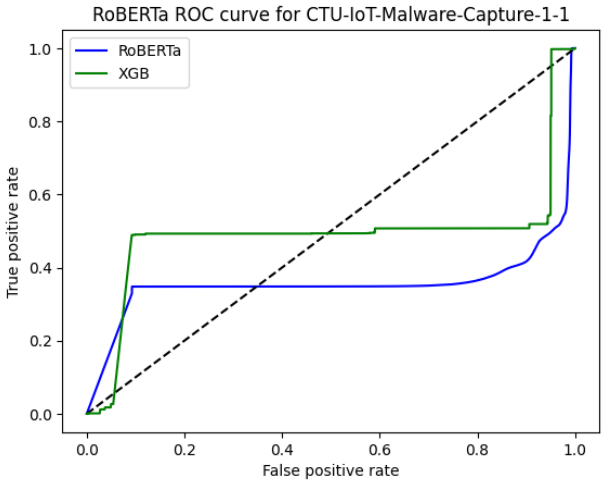
	RoBERTa	XGB
Accuracy	0.59604	0.68229
Precision	0.80290	0.85915
Recall	0.32424	0.48552
F1-score	0.46193	0.62043

Label Counts

	Actual	RoBERTa	XGB
1	539473	790889	703882
0	469275	217859	304866





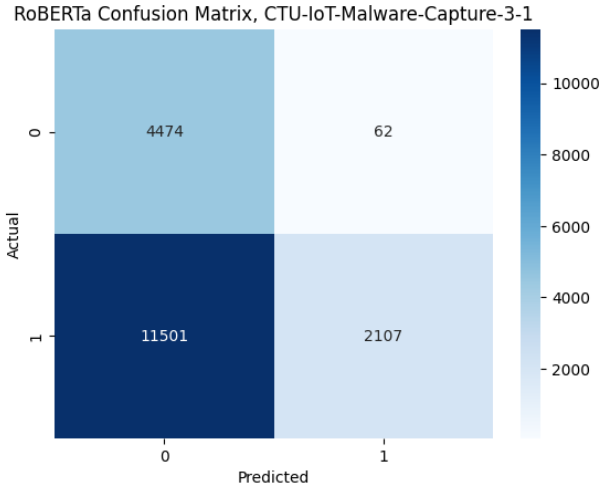


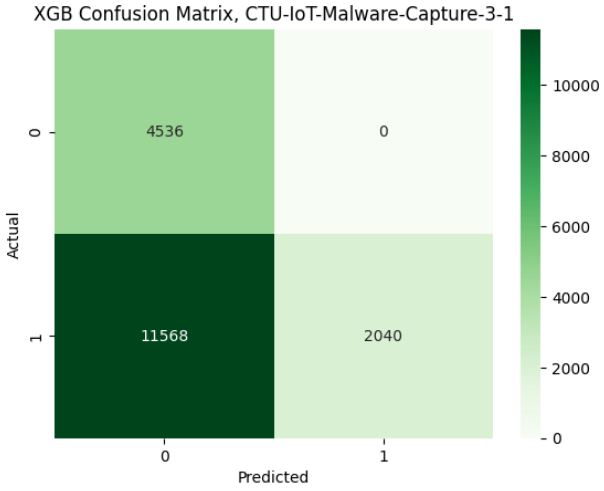
03_Muhstik

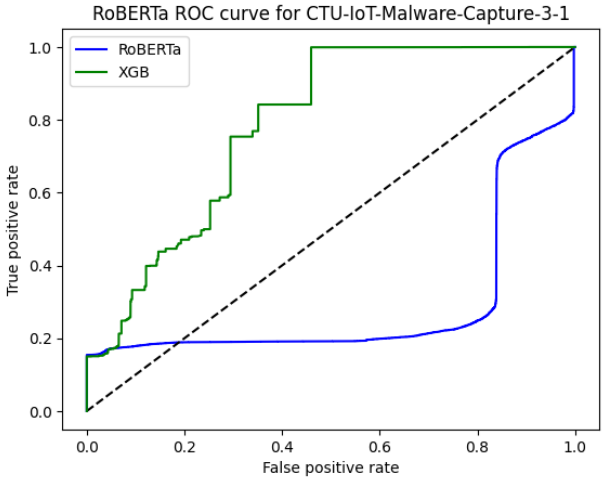
	RoBERTa	XGB
Accuracy	0.36271	0.36243
Precision	0.97142	1.00000
Recall	0.15484	0.14991
F1-score	0.26710	0.26074

Label Counts

	Actual	RoBERTa	XGB
1	13608	15975	16104
0	4536	2169	2040





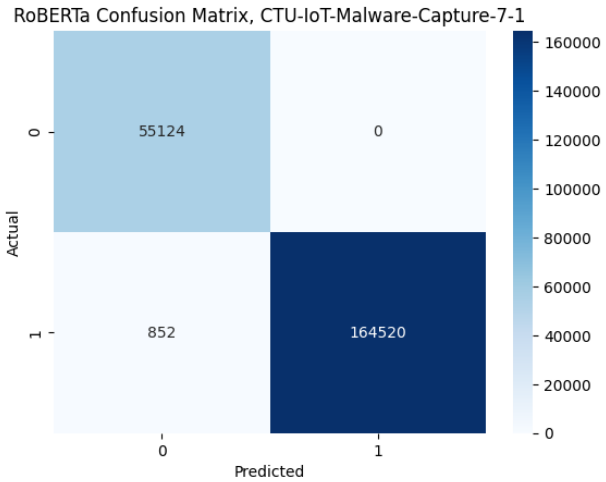


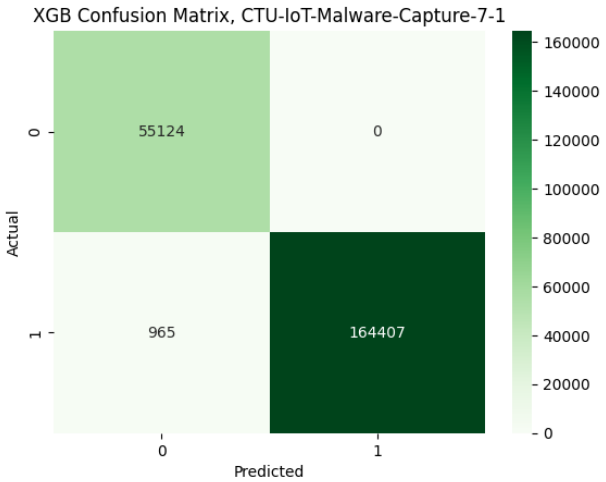
07_Mirai

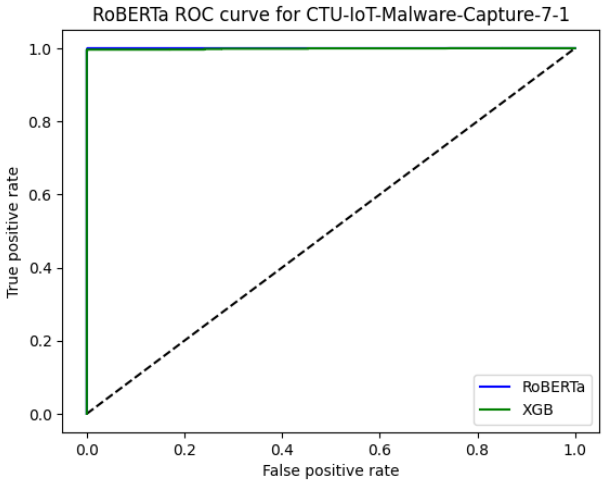
	RoBERTa	XGB
Accuracy	0.99614	0.99562
Precision	1.00000	1.00000
Recall	0.99485	0.99416
F1-score	0.99742	0.99707

Label Counts

[Test Split]	Actual	RoBERTa	XGB
1	165372	164520	164407
0	55124	55976	56089





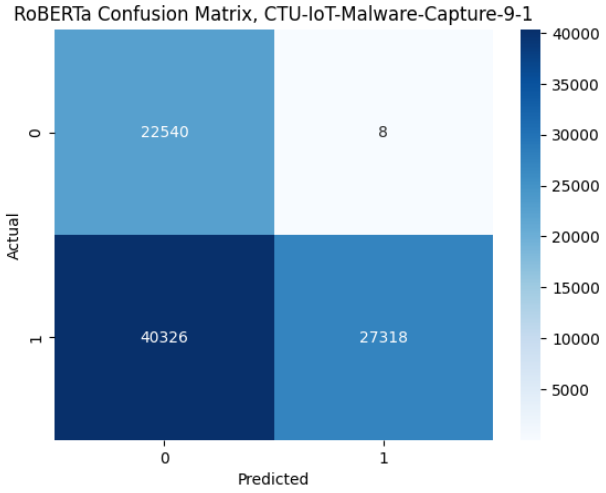


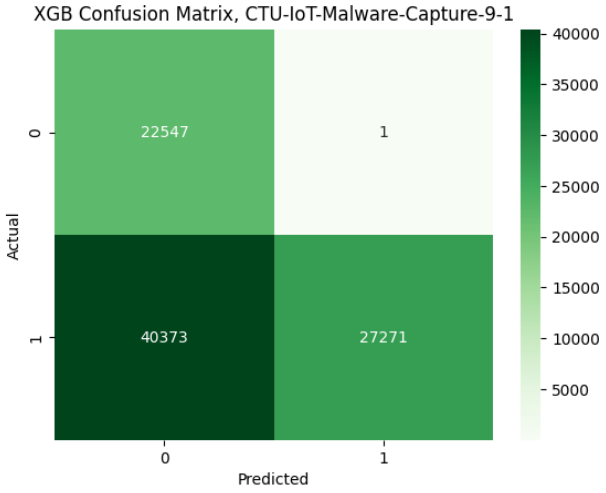
09_Hajime

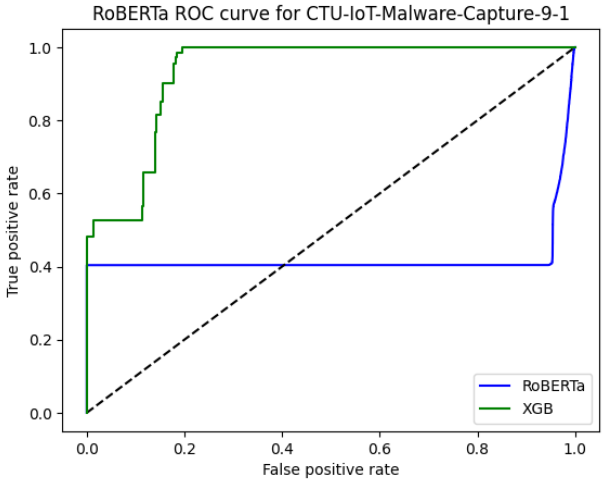
	RoBERTa	XGB
Accuracy	0.55280	0.55235
Precision	0.99971	0.99996
Recall	0.40385	0.40315
F1-score	0.57530	0.57463

Label Counts

	Actual	RoBERTa	XGB
1	67644	62866	62920
0	22548	27326	27272





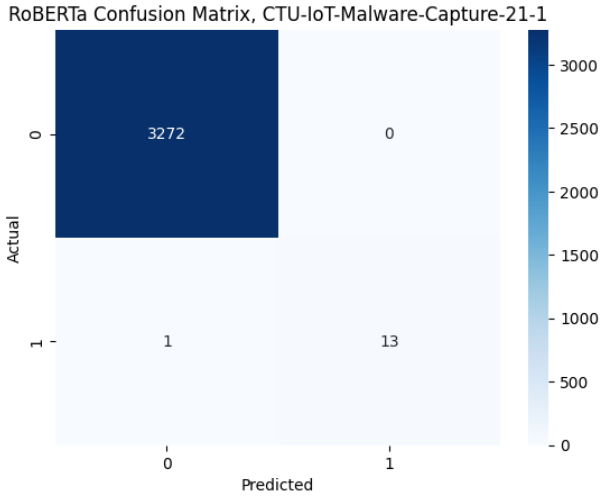


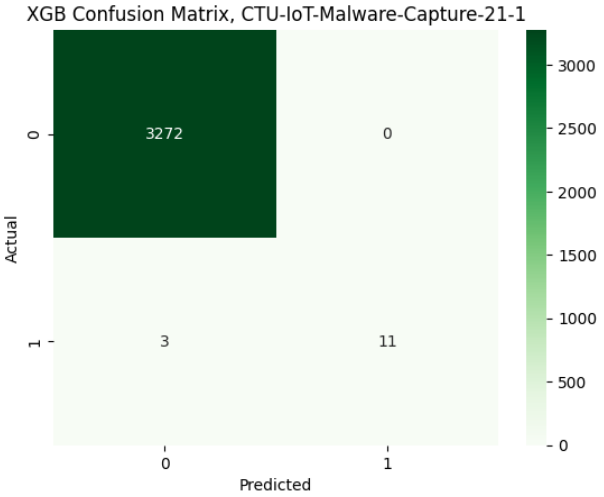
21_Torii

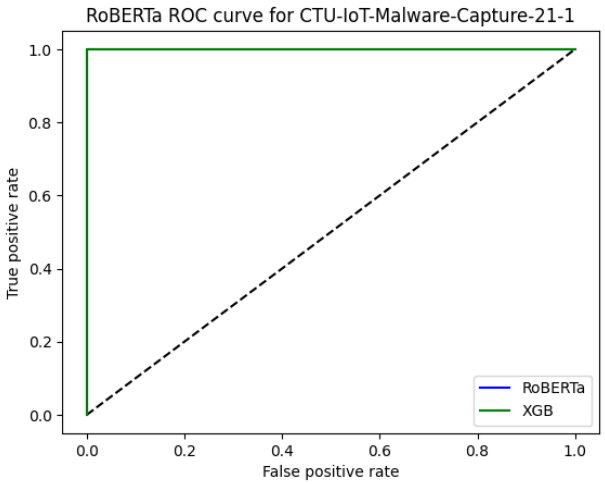
	RoBERTa	XGB
Accuracy	0.99997	0.99909
Precision	1.00000	1.00000
Recall	0.92857	0.78571
F1-score	0.96296	0.88000

Label Counts

	Actual	RoBERTa	XGB
1	3272	3273	3275
0	14	13	11





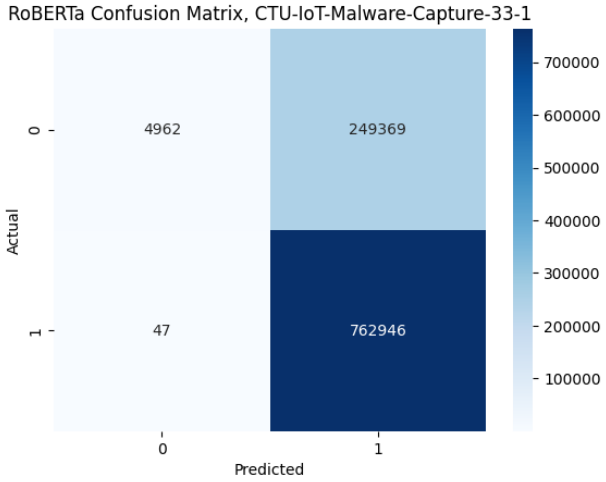


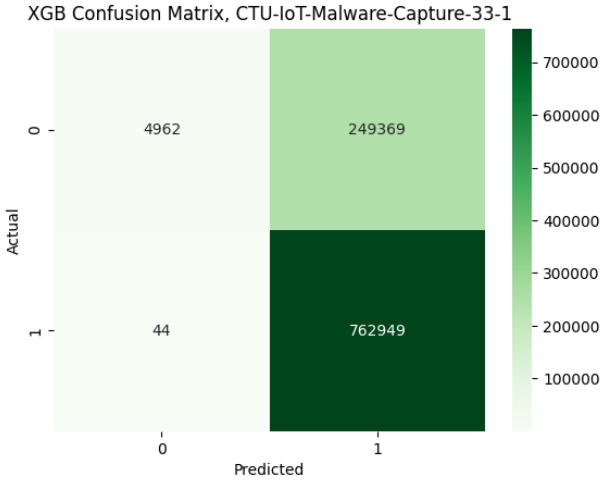
33_Kenjiro

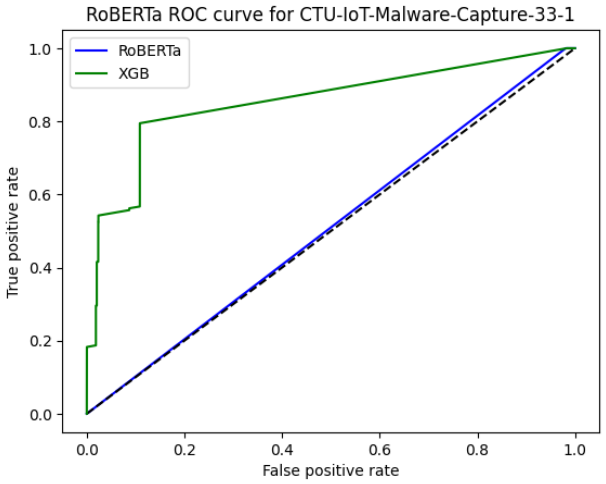
	RoBERTa	XGB
Accuracy	0.75483	0.75483
Precision	0.75366	0.75367
Recall	0.99994	0.99994
F1-score	0.85951	0.85951

Label Counts

	Actual	RoBERTa	XGB
1	762993	1012315	1012318
0	254331	5009	5006





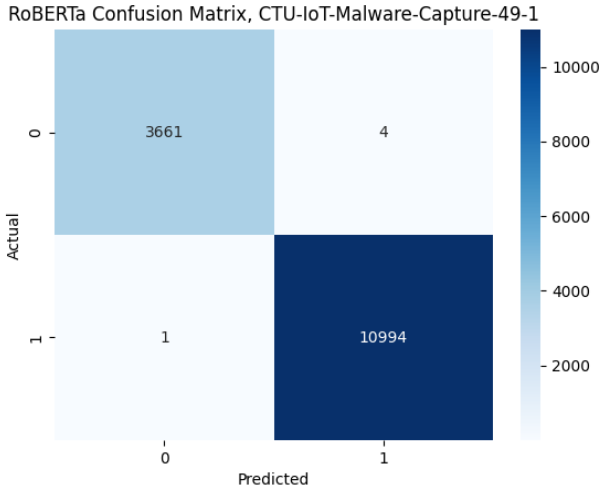


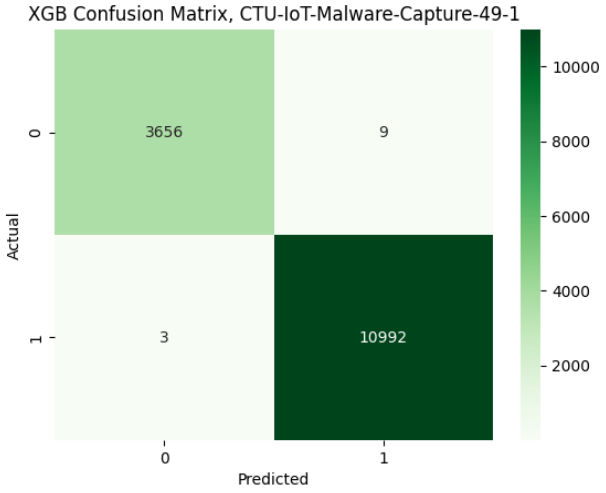
49_Mirai

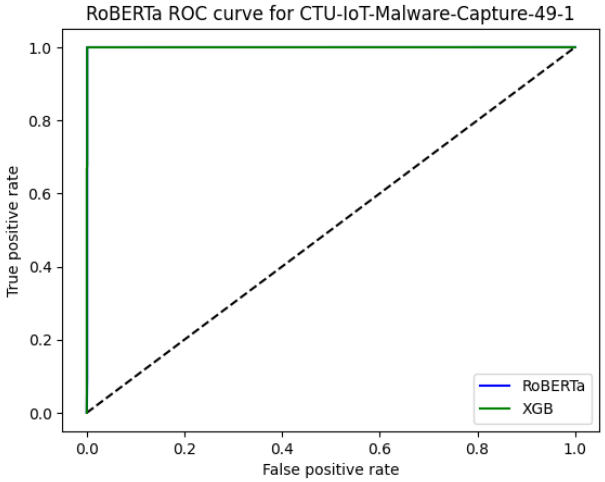
	RoBERTa	XGB
Accuracy	0.99966	0.99918
Precision	0.99964	0.99918
Recall	0.99991	0.99973
F1-score	0.99977	0.99945

Label Counts

	Actual	RoBERTa	XGB
1	10995	10998	11001
0	3665	3662	3659





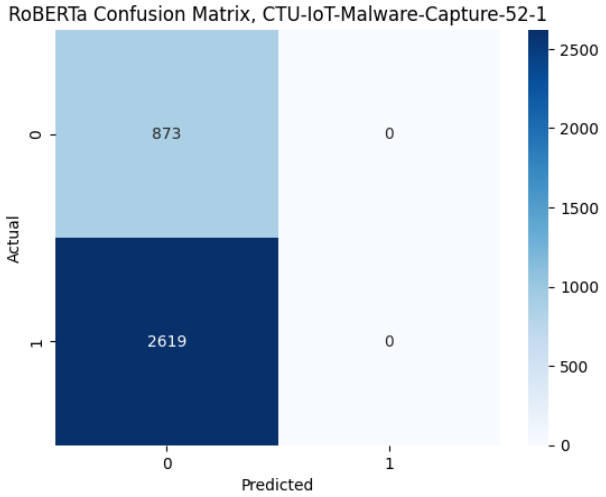


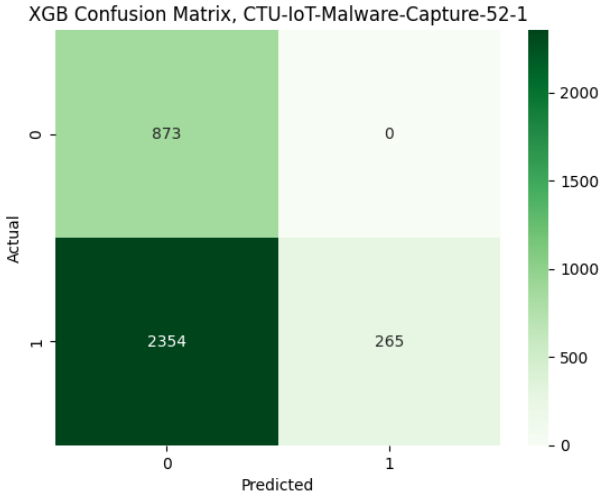
52_Mirai

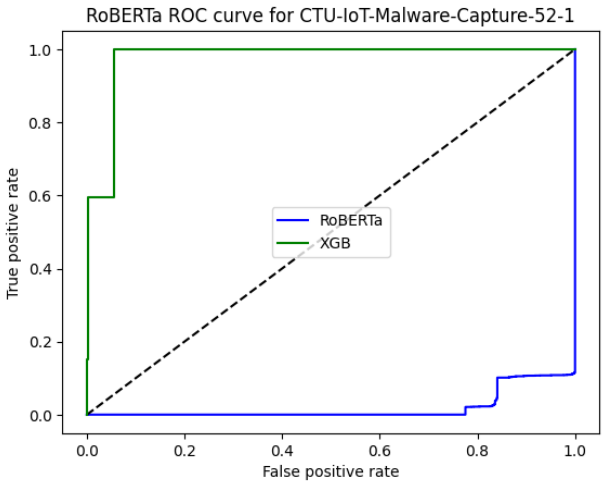
	RoBERTa	XGB
Accuracy	0.25000	0.32589
Precision	0.00000	1.00000
Recall	0.00000	0.10118
F1-score	0.00000	0.18377

Label Counts

	Actual	RoBERTa	XGB
1	2619	0	265
0	873	3492	3227







60_Gagfyt

	RoBERTa	XGB
Accuracy	0.24950	0.82351
Precision	0.00000	0.80950
Recall	0.00000	1.00000
F1-score	0.00000	0.89472

Label Counts

	Actual	RoBERTa	XGB
1	7428	9899	9176
0	2476	5	728