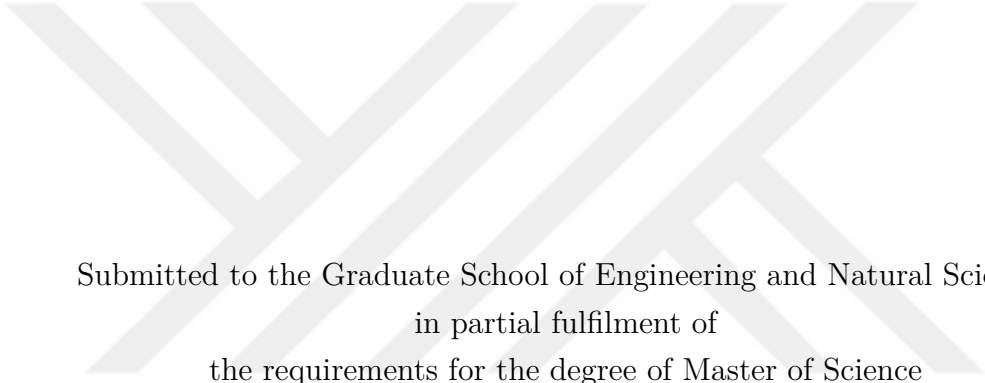


**INTELLIGENT CYBER ATTACK DETECTION USING SOCIAL  
MEDIA POSTS**

by  
MUSTAFA AYDIN



Submitted to the Graduate School of Engineering and Natural Sciences  
in partial fulfilment of  
the requirements for the degree of Master of Science

Sabanci University  
December 2021

**INTELLIGENT CYBER ATTACK DETECTION USING SOCIAL  
MEDIA POSTS**

Approved by:

[Redacted signature]

[Redacted signature]

[Redacted signature]

[Redacted signature]

[Redacted signature]

Date of Approval: December 15, 2021



MUSTAFA AYDIN 2021 ©

All Rights Reserved

## ABSTRACT

### INTELLIGENT CYBER ATTACK DETECTION USING SOCIAL MEDIA POSTS

MUSTAFA AYDIN

Computer Science and Engineering, Master's Thesis, December 2021

Thesis Supervisor: Prof. Albert Levi

Thesis Co-Supervisor: Asst. Prof. Reyyan Yeniterzi

Keywords: cyber security, deep learning, NLP, OSINT

The number of cyber attacks increases every day, so the number of people affected by these attacks is also increasing. For this reason, companies and users need to be aware of the attacks as fast as possible to take precautions and to minimize the loss and effects caused by the attacks. In this thesis, a framework is proposed to detect cyber attacks from Twitter so that entities can act accordingly. The framework consists of two main tasks: tweet classification and information extraction. Two different deep learning based transformers, namely BERT and RoBERTa, are used for our tasks. Two new datasets, one is for binary classification named SUCyber, and the other is for named entity recognition named SUCyberNER, are created. Moreover, an additional dataset from another work is used to evaluate the approaches for the classification. The model that we propose for tweet classification yields average F1-score of 90.1% on two different datasets. Also, the NER model achieves F1-score of 92.29% for the selected tag. In addition, the proposed model has been incorporated into a website that collects and analyzes tweets in real-time to identify DDoS attacks. Finally, this study shows that tweets can be a good source of information to identify ongoing cyber attacks.

## ÖZET

### SOSYAL MEDYA PAYLAŞIMLARI KULLANILARAK AKILLI SİBER SALDIRI TESPİTİ

MUSTAFA AYDIN

Bilgisayar Bilimi ve Mühendisliği, Yüksek Lisans Tezi, Aralık 2021

Tez Danışmanı: Prof. Dr. Albert Levi

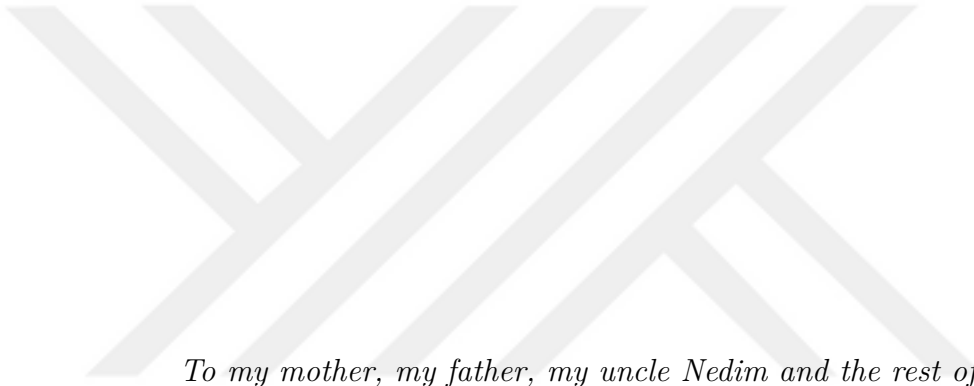
Tez Eş Danışmanı: Dr. Öğr. Üyesi Reyhan Yeniterzi

Anahtar Kelimeler: siber güvenlik, derin öğrenme, doğal dil işleme, açık kaynak istihbaratı

Siber saldırıların sayısı ve doğal olarak bu saldırılardan etkilenen insanların sayısı her geçen gün artmaktadır. Bu nedenle şirketler ve kullanıcılar siber saldırılar sonucu oluşabilecek kayıp ve hasarı en aza indirmek ve önlem alabilmek amacıyla bu saldırılardan olabildiğince çabuk bir şekilde haberdar olmaları gerekir. Bu tezde, siber saldırılardan Twitter paylaşımları kullanılarak haberdar olabilmek için bir çerçeve çalışma sunulmuştur. Bu çerçeve çalışma, tweet sınıflandırması ve bilgi çıkarımı olmak üzere iki ana görevden oluşmaktadır. Bu görevler için derin öğrenme modelleri olan dönüştürücülerden (BERT ve RoBERTa) faydalanılmıştır. Sınıflandırma görevi için SUCyber ismi verilen, varlık ismi tanıma görevi için ise SUCyberNER ismi verilen iki yeni veri seti oluşturulmuştur. Ayrıca başka bir çalışmanın veri seti de sınıflandırma modellerini değerlendirmek amacıyla kullanılmıştır. Sunmuş olduğumuz modelin tweet sınıflandırma performansı iki farklı veri seti üzerinde ortalama %90.1 F1-Skoru olarak ölçülmüştür. Ayrıca, varlık ismi tanıma görevi için sunmuş olduğumuz model seçilen etiket için %92.29 F1-Skoru vermiştir. Tüm bunlara ek olarak, gerçek zamanlı olarak tweet toplayıp geliştirilen model ile analiz eden ve yayınlayan bir websitesi de hayata geçirilmiştir. Sonuç olarak bu çalışma bize tweetler kullanılarak devam eden siber saldırıların belirlenebileceğini göstermiştir.

## ACKNOWLEDGEMENTS

Firstly, I would like to express my gratitude to my thesis supervisor Prof. Albert Levi and my thesis co-supervisor Asst. Prof. Reyhan Yeniterzi. They always guided me from the very beginning to the present and made it possible to complete this thesis with their valuable support. This thesis was prepared during the difficult pandemic that the world was not faced for a long time. Therefore, I would like to thank my advisors again for dedicating their precious time to me and guiding me in this difficult period. I also want to extend my gratitude to the jury members Assoc. Prof. Kamer Kaya, Asst. Prof. Pelin Angin and Asst. Prof. Onur Varol for making their time to participate in my jury and for providing their valuable feedback to this thesis.



*To my mother, my father, my uncle Nedim and the rest of my family.*

## TABLE OF CONTENTS

<b>LIST OF TABLES</b> .....	<b>x</b>
<b>LIST OF FIGURES</b> .....	<b>xi</b>
<b>1. INTRODUCTION</b> .....	<b>1</b>
<b>2. Background Information</b> .....	<b>5</b>
2.1. Cyber Security .....	5
2.2. Deep Learning and Natural Language Processing (NLP) .....	7
<b>3. Related Work</b> .....	<b>9</b>
3.1. Statistical Models, Machine Learning and Classical Neural Networks .	9
3.2. Deep Learning Models .....	11
<b>4. Proposed Approach</b> .....	<b>14</b>
4.1. Tweet Collection .....	15
4.2. Classification .....	16
4.3. Named Entity Recognition .....	18
4.4. Bot Account Detection .....	18
4.5. Example Scenario .....	19
<b>5. Experimental Settings</b> .....	<b>21</b>
5.1. Classification Datasets .....	21
5.1.1. SUCyber .....	21
5.1.2. Second Dataset .....	23
5.2. Named Entity Recognition Dataset (SUCyberNER) .....	24
5.3. Baseline for Classification Task .....	24
5.3.1. CNN Model .....	25
5.3.2. BiLSTM Model .....	26
5.3.3. Word Embeddings .....	26
5.4. Baseline for NER Task .....	27



<b>6. Experiments</b> .....	<b>28</b>
6.1. Tweet Classification .....	28
6.2. Classification Results .....	29
6.2.1. Results for First Dataset (SuCyber) .....	29
6.2.2. Results for Second Dataset .....	31
6.2.3. Results for Combined Dataset .....	32
6.3. Named Entity Recognition and Results.....	35
<b>7. Discussion</b> .....	<b>36</b>
7.1. Tweet Classification.....	36
7.2. Named Entity Recognition.....	39
<b>8. Website</b> .....	<b>42</b>
<b>9. Conclusion and Future Work</b> .....	<b>46</b>
<b>BIBLIOGRAPHY</b> .....	<b>47</b>
<b>APPENDIX A</b> .....	<b>50</b>

## LIST OF TABLES

Table 5.1. Evaluated CNN parameters .....	26
Table 6.1. Classification Results for SUCyber .....	30
Table 6.2. Classification Results for Second Dataset .....	32
Table 6.3. Combined Dataset Results .....	34
Table 6.4. NER Results for Victim Tag .....	35
Table 7.1. Effect of the Training Data Size.....	39

## LIST OF FIGURES

Figure 2.1. DDoS Attack Demonstration by using Botnet .....	6
Figure 4.1. Proposed Framework .....	15
Figure 4.2. Binary Classification .....	17
Figure 4.3. Makeup Tweets for Demonstration .....	19
Figure 4.4. Expected Classification Result for Makeup Tweets.....	20
Figure 4.5. Expected NER Result for Makeup Tweets .....	20
Figure 5.1. Example for Tweet Pre-Processing .....	23
Figure 8.1. Example of Named Entity Recognition and Grouping Tweets .	42
Figure 8.2. Result table from the website for September 10, 2021.....	44
Figure 8.3. Example list of tweet IDs for Yandex from the website for September 10, 2021. ....	45
Figure 8.4. Example tweet selected randomly from the IDs list for Yandex from the website for September 10, 2021.....	45

## LIST OF ABBREVIATIONS

OSINT:	Open-source Intelligence
DNS:	Domain Name System
DDoS:	Distributed Denial-of-Service
NER:	Named Entity Recognition
NLP:	Natural Language Processing
SVM:	Support Vector Machine
CNN:	Convolutional Neural Network
IDCNN:	Iterated Dilated Convolutional Neural Network
RNN:	Recurrent NeuralNetwork
GRU:	Gated Recurrent Unit
BiGRU:	Bidirectional Gated Recurrent Unit
LSTM:	Long-Short Term Memory
BiLSTM:	Bidirectional Long-Short Term Memory
CRF:	Conditional Random Field
LDA:	Latent Dirichlet Allocation
ELMo:	Embeddings from Language Model
BERT:	Bidirectional Encoder Representations
RoBERTa:	Robustly Optimized BERT Pretraining Approach
XLNet:	Generalized Autoregressive Pretraining for Language Understanding
ULMFiT:	Universal Language Model Fine-tuning

## 1. INTRODUCTION

The power and frequency of cyber attacks increase every day. The skills to perform a cyber attack and the resources needed to carry out the attacks became easier for some types of cyber attacks such as DDoS attack. For example, people can rent botnets easily from hackers for very cheap to perform a DDoS attack nowadays. In addition, the Covid-19 pandemic also have a great impact on the increase of the number of cyber attacks because much more people had to use the internet for lots of different tasks for a much longer time (Lallie, Shepherd, Nurse, Erola, Epiphaniou, Maple & Bellekens, 2021). Therefore, the number of possible victims and the number of attack field increased. Thus, detection of the events that are related to cyber security and protection capabilities of entities against cyber attacks should be improved to stay safe. For this purpose, users and security analysts need to follow different information sources and open-source intelligence (OSINT) plays an important role nowadays. Cyber security news, articles, and code repositories are examples of the information source for OSINT. In addition, diversification and expanding the sources can make people and analysts more prepared against attacks so mitigation of the attacks can be more successful, and the damage caused to the user by the cyber attack can be minimized.

People share their ideas about topics or events that affect them on social media platforms such as Twitter. Millions of social media posts are shared about a variety of topics. These posts are called tweets for Twitter and approximately 500 million tweets are shared per day as cited in (Antonakaki, Fragopoulou & Ioannidis, 2021). In this work, the objective is to utilize the regular tweets that are posted by any type of user to save other users against cyber attacks by detecting and processing the tweets. For this objective, the first question is can we classify the tweets automatically as to whether they are related to a cyber security event. The following question is can we extract victim information automatically from the tweets that are classified as relevant. Therefore, different methods were performed and their results were evaluated to find the best approach for the objectives. Lastly, the aim is to combine the best approaches from the findings about our tasks to propose a

framework to be used end-to-end for the detection from Twitter. In brief, we explore the ways and their success rates for using the flow of tweets as a rich seam of information because some portion of them includes important knowledge about events, especially instant ones. For example, a currently unavailable web service that is affected by a DDoS attack can be reported by a user, who cannot access and use it properly, by posting a tweet immediately. Another example can be a tweet that reports a data breach discovered by a user. If users share the data breach on Twitter and if it can be detected by systems automatically, other victims of a cyber attack can be alerted and they can protect their assets.

Cyber attacks can have different purposes like damaging people by stealing assets or making a service unavailable. Regardless of the aim of an attack, a user has a right to learn about the situation as soon as possible. However, cyber attacks are sometimes noticed late or not reported to the user at all. Therefore, any type of cyber attacks should be detected fast and tweets that report the attacks can be used for this purpose. (Naaman, Boase & Lai, 2010) states that only 20% of users of Twitter shares informative posts. Therefore, only a small portion of all tweets include valuable information about the cyber attacks and besides the valuable information, there are many tweets which are unrelated to any cyber attacks. Due to the huge number of tweets posted on Twitter, the classification of whether a tweet reports a cyber attack cannot be handled manually. Therefore, an automated system is needed for the classification and extracting the information about the attack from it. Thus, cyber attacks can be detected and beat fast, the impact of the attack can be reduced and, damage or loss for users can be minimized. Also, datasets about cyber attacks can be constituted for further analysis such as analyzing the attack frequency, observing the attack types to improve security accordingly and detecting the impact of an attack.

In this thesis, a framework is proposed for cyber attack detection from Twitter and information extraction from the detected tweets to reach details of the cyber attack using deep learning for NLP. In other words, the framework consists of two main tasks: classification for tweet detection and Named Entity Recognition (NER) for information extraction. Firstly, collected tweets are classified about its relevance with cyber security events and if a tweet is classified as relevant NER applied to it to extract the victim information. After the NER, we added a detection mechanism against bot accounts to minimize the number of collected tweets that report a fake cyber event. For the main tasks, transfer learning and transformers are evaluated, which were not used in most of the recent studies that have a similar aim in cyber security domain, together with popular deep learning models such as CNN and BiLSTM. In other words, ELMo embeddings which is a word embedding technique,

ULMFiT, BERT, RoBERTa and XLNet are evaluated for the main tasks in this thesis. For the bot account detection phase, we use a machine learning algorithm called Botometer from another study Sayyadiharikandeh, Varol, Yang, Flammini & Menczer (2020).

There is a need for a dataset to show and evaluate the results of the work. However, there is no ready-to-use dataset to work on it and get results to compare your results with other works because of the Twitter privacy policy. People can only share the IDs of tweets and if a tweet is deleted or the user who posted the tweet is no longer exists, the tweet cannot be reachable. Therefore, authors create their own dataset and either do not share the dataset or share only IDs of tweets in the dataset in most of the works. At this point, transfer learning can play an important role because it can provide to reach better performance with small datasets. Due to the limitation, we also created a new dataset by selecting Distributed Denial-of-Service (DDoS) as an example attack type. The DDoS attack is performed to disable web services by exceeding the request capacity of the resource, so it affects all user who wants to use the service. The main reason for the selection of the DDoS attack is that it is a mass attack. It means that an attack can possibly affect a great number of people according to the number of users of a victim. In addition, it is one of the most preferred attack types, as it can indirectly harm many people who is user of main victim by targeting a main victim. Also, the power and frequency of the attack are increasing day by day. According to the statistics, there were 5.4 million DDoS attacks that occurred, that is a new record, in the first half of 2021 and the number was approximately 4.83 million in the same period of 2020 (Netscout, 2021). The dataset was created by collecting and labeling tweets posted about real DDoS attacks in the corresponding attack dates. Finally, the dataset includes 4013 positives and 2019 negative tweets. In this work, a positive tweet means that it is relevant to a cyber security event, negative tweet means that it is not relevant. In addition, we reached a dataset that consists of pre-processed tweets of another work (Yagcioglu, Seyfioglu, Citamak, Bardak, Guldamlasioglu, Yuksel & Tatli, 2019), called as second dataset in this work, and used it for evaluation purposes. The second dataset is constituted by security experts and it includes 843 positives and 1474 negative tweets about different cyber security events. For the classification, we used these two datasets separately. Lastly, 1501 positive tweets from our dataset were randomly selected and manually annotated with victim entity to evaluate our NER approach. More details for the datasets can be seen in Chapter 5.

The evaluation results shows that our approach reaches 98.89% F1-score for the classification on our dataset. Also, the F1-score for the classification with our approach on the second dataset is 81.30%, and it is 72.02% with the approach of study

(Yagcioglu et al., 2019) which is the owner of the second dataset. For the NER task, the F1-score for the victim entity is 92.29%, and it is 86.65% by using BiLSTM-CRF model (Dionísio, Alves, Ferreira & Bessani, 2019).

Contributions of the paper can be summarized as follows:

- A new dataset which consists of tweets about DDoS attacks was created by manual labeling for classification.
- A dataset for named entity recognition was created by applying manual annotation.
- A new framework based on transformers is proposed for cyber event detection from Twitter.
- F1-score of baseline models based on previous studies are outperformed for both classification and NER tasks.
- The F1-score of another study on their dataset is improved with our approaches for binary classification.
- Transfer learning and transformers which were not used in most of the recent studies in cyber security domain are evaluated for the tasks.
- The proposed framework includes bot account detection phase, that does not exist in examined relevant works, so the detection is more accurate against unreal claims for cyber security events shared in Twitter.
- Results of the proposed approach are shared in real-time via a website.

The rest of the thesis is structured as follows: in chapter 2 we provide background information to make rest of the paper more clear. In chapter 3, related works are reviewed and discussed. In the chapter 4, the proposed approach is explained in detail. The experimental settings and details of experiments together with results and analysis are shared in chapter 5 and 6, respectively. In chapter 7, there is a discussion about the study in general. The details of the website and an example result can be found in Chapter 8. The last chapter is the Chapter 9 that is the conclusion of this thesis.



## 2. Background Information

This study is related to different computer science subjects such as cyber security, deep learning and, NLP. We aim to detect cyber attacks as fast as possible from social media by utilizing deep learning for NLP to respond the attacks fast and minimize its effect. In this chapter, background information to understand better the work with all details can be found for the subjects that exists in this study.

### 2.1 Cyber Security

Attackers aim to harm people by disabling some services, stealing assets, and destroying resources by applying different types of cyber attacks. On the other hand, people should be aware of the attacks and follow current security recommendations. Also, people who are responsible for the security of a product or company should be prepared against possible attacks by examining past and recent attacks.

**Hackers** There are different types of hackers but black hat and white hat ones are the most well-known types. The black one is a person who aims to harm people by performing cyber attacks. On the contrary, white hat hackers should be one step ahead of black hats to prevent their possible attacks. They are also called cyber security experts in this study.

**Open-source intelligence (OSINT)** It is a technique to deduce meaning from public data by collecting and processing it. It can be used for any field but it is related with cyber security in this study. Cyber security experts should follow different sources to be prepared for any attacks and OSINT is a one way to follow cyber security events.

**Distributed Denial of Service (DDoS)** It is one of the most frequent and effective types of cyber security attack. Attackers aim to take down a web service by sending too many requests to the server to exceed the capacity of server. Thus, products who are running under the attacked service are affected from the attack so the number of victims can be huge by aiming single target. For example, a series of DDoS attacks were targeted to Dyn DNS, which operates mapping domain names with IP addresses, in recent history. In this attack, more than 70 websites that uses Dyn as a DNS provider like Amazon, PayPal and Twitter were affected. It became the reason for preference because it causes more victim with less effort. Mostly, botnets are used for DDoS attacks to make a large number of web requests.

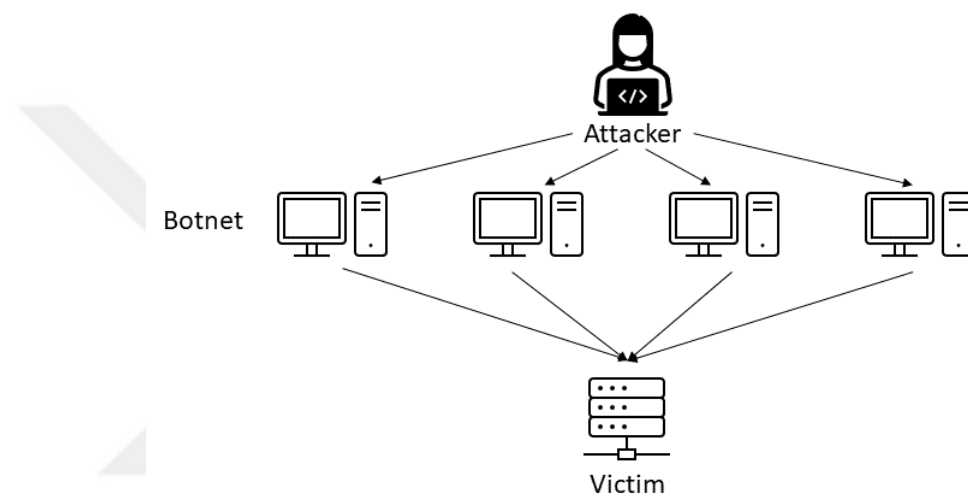


Figure 2.1 DDoS Attack Demonstration by using Botnet

**Botnet** Any type of device that is connected with the internet like IP cameras and affected from malware and managed by hackers. Botnets can expand their bot army by spreading malware to more devices so the total power increases. These devices can be used for sending millions of web requests to perform DDoS attacks. One of the well-known botnet is called Mirai that spread mostly in IoT devices via malware (Örs, Aydın, Boğatarkan & Levi, 2021). It was also the source of the DDoS attack to Dyn DNS that was mentioned above.

**Malware** It is a type of software that aims to damage computer systems. Worms and trojans can be example types of malware.

**Phishing** It is a technique to steal credentials or data of users by cloning a website as it is and presenting it as the real website. The user thinks that he or she interacts with the real website, but the website is fake and the entered data is obtained by the attackers.

**Data Breach** This is a term that is used to explain leakage of data that should be confidential and stored safely. It can be a login credential or valuable assets like credit card information. The breach can happen as a result of cyber attack. Also, the reason can be a fault of an entity that is responsible for data storage, or it can be the result of a software error. The importance of data breach depends on the source of the data and protection techniques like hashing if applied to the stolen data. However, if the stolen data includes login credentials it can be very dangerous because people tend to use the same credentials for many websites so a leakage can cause much loss. Therefore, a data breach should be detected and notified as fast as possible so users can take action and minimize their loss.

## 2.2 Deep Learning and Natural Language Processing (NLP)

In order to detect events related with cyber security automatically, we needed to process text data that are tweets in this study. For this purpose, we use different NLP techniques and deep learning models. We evaluate different deep learning algorithms in this study to achieve the best performing model and background information about them can be found in this section. Also, more information is shared in the following related sections.

**Deep Learning** It is a specialized group of machine learning algorithms for several tasks such as computer vision and natural language processing. The method is based on artificial neural networks that mimic biological human brains so there are layers of network in these models. There are well-known deep learning algorithms for different tasks such as Convolutional Neural Network (CNN) and Recurrent Neural Networks (RNN).

**CNN** It is a deep learning model which is mostly used for computer vision tasks but later it is also used for NLP so it is evaluated in this study for base model selection. There are different types of layers that are added one after another to build the model.

**BiLSTM** It is also one of the most popular deep learning models that consists of two LSTM models that are processing the input in opposite direction so the context of a text can be learnt better with this model.

**Transformers** It is a deep learning model, that consists of encoders and decoders, which is one of the most successful approaches today for different tasks so they are evaluated in this study in detail. It utilizes the mechanism of attention that provides to manage importance in a different parts of input data. There are very popular pre-trained transformers so they can be used on different datasets by applying fine-tuning to the models and state-of-the-art results can be obtained.

**Bidirectional Encoder Representations (BERT)** It is one of the most popular transformers that was pre-trained by Google researchers for the next sentence prediction and language modeling task of NLP on huge datasets in 2018. It consists of encoded layers and self-attention heads.

**Robustly Optimized BERT Pretraining Approach (RoBERTa)** After the breakthrough of BERT, researchers aimed to achieve better approaches to reach state-of-the-art results. RoBERTa was created by Facebook researchers in 2019. They changed hyperparameters and removed the next-sentence prediction part that exists in BERT to achieve better training performance. Therefore, they trained the model on 10 times higher sized dataset than BERT so they were able to achieve better scores in different tasks.

**Generalized Autoregressive Pretraining for Language Understanding (XLNET)** The model uses a different language modelling called permutation to achieve better results than BERT. Also, they pre-trained their models on a larger dataset with more computational power than BERT to achieve a better results in NLP tasks. Therefore, it is evaluated for our task in this study.

**Transfer Learning** It is a methodology to utilize the power of a model for different task or data. As mentioned above, there are very powerful pre-trained transformers so their power can be used on smaller datasets with fine-tuning the models. Therefore, better results can be achieved with them rather than training a model from scratch.

### 3. Related Work

In this chapter, existing studies that include text detection for events related to cyber security from social media posts and information extraction about its detail are reviewed. In other words, these works aim to use social media as an OSINT source for the cyber security aspect. The studies are grouped in two subgroup according to their methodologies.

#### 3.1 Statistical Models, Machine Learning and Classical Neural Networks

In this section, previous works that used statistical models, machine learning and classical neural networks for the detection of tweets are shared. Firstly, (Ritter, Wright, Casey & Mitchell, 2015) proposed a weakly supervised approach for cyber security event detection. The advantage of the approach is that it can be constituted by using only a small number of positive tweets. Also, their tweet collection approach is very useful for other works and it is used for our work to create the dataset. They did not assume unlabeled tweets as negative as previous works rather they create a learning approach that is based on statistical approaches to label them. It is a useful approach in terms of dataset constitution but its success is limited and the approach can perform worse than more recent deep learning approaches in most cases. In another study, (Le Sceller, Karbab, Debbabi & Iqbal, 2017) proposed a framework to detect cyber security events automatically like our aim, but their method is depending on extraction of keywords from tweets for clustering by using algorithms based on statistical approaches from other studies.

Alternatively, (Chambers, Fry & McMasters, 2018) uses the same approach in (Ritter et al., 2015) to create a dataset and they do not share their dataset. Similar to our work, DDoS is selected as example attack type for the dataset. Differently, manual selection for collected tweets about its relevancy was not applied. The authors made

an assumption for the relevancy of tweets according to the date of tweets. However, this approach is too generic and it decreases the quality of the dataset. In our work, the dataset was handpicked so it is expected to be more accurate because there is no assumption. Other than that, their study proposed two learning frameworks which consist of a feed-forward neural network and a partially labeled Latent Dirichlet Allocation (LDA) statistical model similarly with (Ritter et al., 2015) for cyber attack detection. Although these are well-known and statistics-based approaches, in most of the cases their performances are lower than the deep learning models that allow us to obtain state-of-the-art results at present in many of the NLP tasks. Different from the other studies, their work also aimed to analyze people’s behaviors during the attack from tweets and it is the particular aspect of their study.

In another work, (Wang, Al-Rubaie, Clarke & Davies, 2017) proposed a warning system for real-time Twitter traffic by using altered version of LDA model and they called it tweet-LDA. Similar to the neural network of (Chambers et al., 2018), (R., Alazab, Jolfaei, K.P. & Poornachandran, 2019) proposed a classical deep neural network that consists of 3 layers. The main difference is the study focuses on Ransomware as example cyber attack type. The authors created a new dataset, that is not public, with 214,463 tweets about ransomware attacks. Their dataset is huge in comparison with the general trending sizes of similar works but its context and quality is a question point. Also their model is a weaker model if we compare it with the well known deep learning models like CNN, BiLSTM or transformers. If the models used in our study were evaluated, they could get much better results depending on the quality of their dataset.

Alternatively, (Ural & Acarturk, 2021) proposed a similar detection mechanism but they only focused on Turkish tweets. In their approach, they used a method based on keyword frequencies called Term Frequency - Inverse Term Frequency. From previous tweets that report any cyber security events, they created a word vector with the mentioned method. Different from other studies, they also use a Turkish newspaper as another source of texts. After the text collection, they processed the text and then applied NER to extract information by using pre-defined vectors of string about possible victim entities. Lastly, they planned to share their results in real-time like us but they did not implement this idea. In another recent study, (Riebe, Wirth, Bayer, Kühn, Kaufhold, Knauthe, Guthe & Reuter, 2021) proposed a system for alerting about cyber security events. They used an existing approach for classification from another work (Habdank, Rodehutsors & Koch, 2017) that compared machine learning models like random forest and support vector classifier with a simple neural network. As a result, random forest gave the best result for their case. Therefore, (Riebe et al., 2021) evaluated slightly different alternatives of

random forest with a very classical model of k neighbors classifier. Consequently, they only focused on machine learning models, but more advanced approaches like deep learning models could be given a chance, especially in the environment where they give state-of-the-arts result in many NLP tasks.

### 3.2 Deep Learning Models

In addition to the statistical models, machine learning, and classical neural network architectures, most of the works use popular deep learning models such as CNN and Recurrent Neural Network (RNN).

There are papers that focuses on only classification task for cyber security related tweets. In our work, we also have the NER task together with bot account detection to minimize false reports. Firstly, (Yagcioglu et al., 2019) aimed to classify tweets about cyber security events so 2000 tweets about the events were collected, and manually annotated by cyber security experts. Also, the dataset that consists of pre-processed tweets is public thanks to the authors. Since the dataset constitution was done by domain experts, and there is no obvious bias in the tweet collection approach, we decided to use this dataset along with our dataset. In their approach, they used CNN and LSTM models for the classification similar with most of the works, but the main difference is that the operations done to tweets. They combined different embeddings and created task-specific features that were not done by another paper. Thus, they improved performance on existing LSTM and CNN approaches but the models stayed same only other components were changed. Therefore, improvement is dependent to the power of the models after a point. In another study, (K, Balakrishna, R & KP, 2020) focuses on the same task with different deep learning models. The other difference is that there is a multi-class classification for event type detection along with binary classification. For these tasks, the study proposed an approach that consists of Convolutional Neural Network (CNN) and Gated Recurrent Unit (GRU) models. They used the dataset of another work (Behzadan, Aguirre, Bose & Hsu, 2018). However, their approach does not work as good as multi-class classification. For binary classification, they achieve F1-score of 77.8% but it is 89.3% for the multi-class classification.

For the single NER task, successful approaches include only the Conditional Random Fields (CRFs) layer or the CRF layer after the LSTM model in general. However,

there can be better approaches in specific domains such as cyber security. Therefore, there are studies that only focus on NER tasks for this domain. (K, S, R & KP, 2020) proposes a model which consists of BiGRU, CNN, and CRF. They evaluated their approach with automatically annotated text about cyber security and showed that it performs better than a single CRF layer and LSTM together with the CRF model. However, we constituted a dataset for the NER task by annotating tweets manually and we evaluate transformers that are more successful in most of the NER tasks than the approaches in their study. In another study, (Tikhomirov, Loukachevitch, Sirotina & Dobrov, 2020) used BERT for NER task in cyber security domain for mainly Russian texts. This is one of the rare studies which uses BERT for NER in this domain. Finally, they showed that BERT works better than CRF for the task in cyber security texts and we evaluated it together with RoBERTa and XLNet in our study for the NER.

There are also studies that consist of the same tasks of our work: data collection, classification, and NER. However, none of them includes the extra control for bot account detection as our study. Firstly, (Dionísio et al., 2019) proposed a processing pipeline that consists of the 3 tasks. They created a dataset for this purpose, and it consists of tweets posted by pre-defined sets of accounts. The main difference is that they aimed to detect cyber threats, not cyber events. It is important to detect these threats before they are exploited. However, they can only be detected and shared by people who are related with cyber security in most cases. Moreover, explaining the threat will include a number of technical terms. Therefore, the amount of tweets and the number of people who will understand and share the information is very limited which can also be seen from the dataset of the study. However, our work aims to detect cyber events from any type of user whether it is related with cyber security or not is not important. Also, there are more people that share the cyber event, so it increases the detection rate and speed. Therefore, the topic is not exactly the same but their approach is useful. They implemented Convolutional Neural Network (CNN) for tweet classification and Bidirectional Long Short-Term Memory (BiLSTM) for the NER task. The authors also aimed to improve their work by changing their framework in another study (Dionísio, Alves, Ferreira & Bessani, 2020). For the tasks, the same deep learning models stay as it is. However, they combined the tasks for creating a multi-task model and simplified the framework. According to their evaluations, the new design achieves better performance for classification but NER performance stays the same as their previous work. Similarly, (Fang, Gao, Liu & Huang, 2020) proposed a framework for the exact same aim but the framework and dataset is different. They created a dataset by collecting tweets about cyber security events from random users for positive tweets. Also,



they collected tweets from persons who are not related with cyber security such as Donald Trump. The IDs of tweets in the dataset are shared without labeling, so the dataset is not useful. Moreover, their approach for collecting negative tweets decreased the power of dataset. The difference between the context of negative and positive tweets is huge, in other words the difference between classes is too distinct. In their framework, they used BiLSTM for classification and BiLSTM together with Iterated Dilated CNN (IDCNN) for the NER task. The main difference between the paper from the others is that they used an additional layer that consists of LDA at the beginning. LDA increases performance but it is related with the power of the dataset as mentioned above. Similarly with the previous work (Dionísio et al., 2020), the effect of multi-task learning design is also evaluated in this work and they also achieved better results than single tasks.

As can be seen in this section, recent studies mostly used deep learning models like CNN and RNNs. Only the last paper that is reviewed in this section about the single NER task uses BERT. However, transfer learning and transformers are provided to achieve state-of-the-art results in many NLP tasks. Therefore, we evaluated different transformers for the classification and NER tasks in this thesis.

## 4. Proposed Approach

In this thesis, we aim to detect cyber security events from tweets. For this purpose, firstly tweets are needed to be classified whether it is related with a cyber event or not. Also, the NER task is needed to be performed to extract information such as the victim of a cyber attack from the positive tweets. In this way, victims and users of them can be alerted to minimize the damage of the attack. Also, detected cyber attacks can be stored in a dataset to share them and they can be used to analyze the details of attacks against a victim. For this purpose, we store positive tweets by combining them according to their victim information for everyday and these are shared in a web page to show the performance of our approach in the real environment. In addition to these, a bot account detection mechanism for Twitter from another work (Sayyadiharikandeh et al., 2020) is integrated to end of the our framework. The aims of the bot detection module are preventing spoofing our approach with untrue posts and preventing to collect posts that include fake threats to scare a victim with bot accounts. Lastly, setting a threshold for the number of tweets posted for a victim is also useful to minimize the false-positive detection for mass attacks. The reason is there are lots of victims in the mass attacks and expecting more tweets is a reasonable expectation. However, for a victim with a small number of users, the threshold should be carefully adjusted to prevent the miss of detecting a real cyber attack. Already, collecting all posted tweets is not possible due to the limitations of Twitter API so we cannot catch all tweets. Moreover, types of cyber attacks and the possible popularity of a victim also need to be considered for setting the threshold.

For the classification and the NER, we use transformers that provide us to achieve better results than the other approaches. As mentioned in Chapter 3, although transformers are powerful, they are not used in many current studies that have a similar aim as this work. Therefore, it is useful to evaluate transformers in this study which provides to achieve best results for many studies in different domains. For example, XLNet (Yang, Dai, Yang, Carbonell, Salakhutdinov & Le, 2020) provided to achieve state-of-the-art results for question answering task on 'Quora Question

Pairs' and 'SQuAD2.0 dev' dataset. Another example is BERT (Devlin, Chang, Lee & Toutanova, 2019) which yielded the best result in Amazon Review benchmarks.

The framework of the proposed approach can be seen in Figure 4.1 and can be examined in upcoming sections in detail.

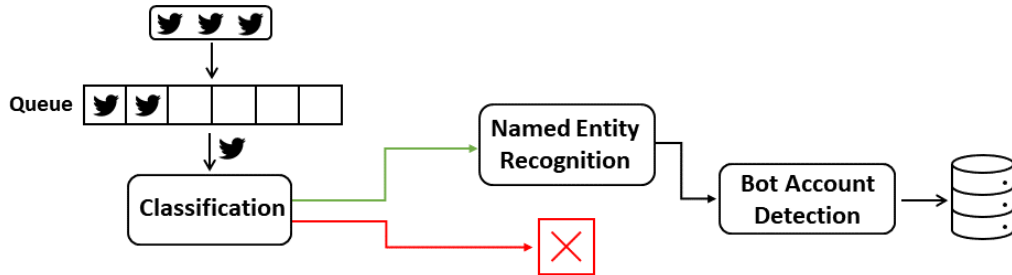


Figure 4.1 Proposed Framework

## 4.1 Tweet Collection

Tweets are collected using Twitter Streaming API in real-time. It is not possible to collect all tweets posted because of limitations of Twitter so Filtered stream <sup>1</sup> is used. In this API, we need to specify a rule for specifying details of the tweet collection. For example, the keyword "DDoS" should be added to the rule to collect tweets about DDoS attacks. Thus, tweets that are more related with cyber security events are expected to be collected. Then, collected tweets are stored in a queue to be processed in order for the tweet collection process to continue independently from other processes and not be interrupted. It is described before the classification in Figure 4.1. An example rule for collecting tweets for a DDoS attack is as follows:

***"DDoS lang:en -is:retweet"***

In the rule above, "DDoS" is the keyword that should be included from tweets to be collected. We also aim to collect English tweets so "lang:en" provides this feature. Lastly, we do not want to collect the same tweet more than once to prevent inconsistency in statistics about an event and it is obtained by adding "-is:retweet" to the rule. Consequently, we collect English tweets about DDoS attacks.

<sup>1</sup><https://developer.twitter.com/en/docs/twitter-api/tweets/filtered-stream>

## 4.2 Classification

Millions of posts are shared every day on Twitter, so there is a need for a classification system to distinguish whether a tweet is related with a cyber security event or not. In this work, baseline models that are CNN and BiLSTM, ELMo embeddings, and ULMFiT, as a transfer learning method, are evaluated. Moreover, we evaluated transformers such as BERT, RoBERTa, and XLNet in the experiments for the binary classification because they are quite successful in a variety of NLP tasks and text classification is one of them. The details of the approaches are as follows:

**ELMo Embedding** This technique provides us to have different word embeddings for the same word in different contexts, unlike the traditional approaches. It also played a role to achieve state-of-the-art results in different NLP tasks in previous works.

**ULMFiT** This is one of the most successful transfer learning techniques in NLP. We fine-tuned the pre-trained language model <sup>2</sup> on our datasets and fine-tuned the classifier <sup>2</sup>. In this way, it is used for binary classification with our datasets.

**Transformers** BERT is a bi-directional transformer that can be fine-tuned for many NLP tasks and it is used for binary classification in this work. It outperformed the results of a variety of NLP tasks and provided to achieve state-of-the-art results. After the breakthrough with BERT in NLP, different studies performed to develop BERT such as RoBERTa were created. In order to increase the performance of BERT, modifications are made, and training was done by using much larger data than BERT. Consequently, improvements on different NLP tasks are achieved so we also decided to evaluate RoBERTa. In addition to these, XLNet model is evaluated which also aimed to increase the performance of BERT. The model was trained on larger data with more computational power. Also, a different language modeling that is based on the prediction of all tokens in random order is used.

As a result, RoBERTa is selected for the classification task to use in the framework. The details of RoBERTa model that is used in here can be found in Appendix A and the evaluation results that is the reason of using RoBERTa model in the framework can be examined in Chapter 6. Also, architecture of the classification

---

<sup>2</sup><https://github.com/fastai/fastai>

can be observed in Figure 4.2. Lastly, explanations of the parts in the architecture can be found at the end of the section.

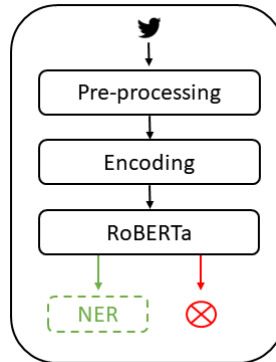


Figure 4.2 Binary Classification

**Pre-processing** Tweets are pre-processed to prepare them ready to use in models. Also, details that do not benefit the model’s better learning are eliminated. The same pre-processing operations applied as in the experiment settings also applied to the proposed approach. More details about the process can be found in Chapter 5.

**Encoding** Tweets are tokenized and prepared for the deep learning models. In other words, expected inputs for the RoBERTa model (Liu, Ott, Goyal, Du, Joshi, Chen, Levy, Lewis, Zettlemoyer & Stoyanov, 2019) that are token ids and attention masks are obtained in this phase. Briefly, tweets are converted to inputs for the model in this stage.

**RoBERTa** RoBERTa (Liu et al., 2019) is a model that was created by modifying BERT (Devlin et al., 2019). The Next Sentence Prediction task that exists in BERT is removed and a new masking method that is called dynamic was introduced. The model is used for binary classification because it gave the best performance in our evaluations.

At the end of the classification, positive tweets are forwarded to the NER model for information extraction and irrelevant tweets are discarded. The details of the NER can be found in the next section.

### 4.3 Named Entity Recognition

There is a need for information extraction from positive tweets to learn the details about of a cyber attack. BERT is used for the NER task because of similar reasons with the classification decision. Briefly, BERT is quite successful in many NLP tasks which include NER and it also gave the best result in our experiments. Fine-tuned BERT model on SuCyberNER dataset from our experiments is used in the framework. The details of the BERT model that is used for NER in the framework can be found in Chapter 5 and the evaluation results that is the reason of using BERT model for the NER can be examined in Chapter 6. At the end of the NER task, if a tweet contains victim information and if it is successfully extracted from the tweet, we perform a control for bot account for the owner of the tweet.

### 4.4 Bot Account Detection

After the NER model, we aim to detect whether the sender account of a tweet is a bot account or not. The aim of the detection is to minimize the detection of untrue claims that are created by using bot accounts about a cyber event. For this purpose, Botometer (Sayyadiharikandeh et al., 2020), which is a machine learning approach, is used. It calculates 6 different scores for different types of behaviors that exist in bot accounts. Also, the API of Botometer (Sayyadiharikandeh et al., 2020) gives a probabilistic score called Complete Automation Probability (CAP) based on the 6 scores and it can be used for bot account classification. Since the CAP score is a probability score, we do not have a direct output for the decision of a bot account. The score indicates what is the probability of an account is labeled as a bot account with calculated 6 different bot scores. The authors of the model suggest using the CAP score for the classification and we accept an account that has CAP score of 95% and more as bot account as suggested by the authors. If the bot score is available for a tweet and if the sender of the tweet is labeled as bot, we do not count this tweet for the extracted victim. Although the tweet is not counted in terms of threshold, it is stored and it can be used for analysis. For example, the data can be used for analyzing the percentage of bot accounts in cyber attack related posts and the frequency of untrue claims for threatening a victim.

At the end of the framework, tweets are stored by grouping them according to the victim information day by day. In this way, the victim and its users can be alerted to minimize the damage of the attack if the number of tweets are more than a threshold. Briefly, people can possibly save other people with their ordinary posts about an instant event that is related to cyber security in our case. An example scenario for the processing of the framework can be seen in Section 4.5.

## 4.5 Example Scenario

An example scenario is constituted to demonstrate the working of proposed approach. In this example, suppose all tweets are shared on the same day and the owner of the tweets is not a bot account. Also, the victim of the cyber attack is Sabanci University and the type of the cyber event is a data breach. Suppose there is a data breach that occurred in the database of Sabanci University and it is realized from few number of users by seeing their correct credentials on the internet. Suppose the following makeup tweets in Figure 4.3 are posted at that time.



Figure 4.3 Makeup Tweets for Demonstration

There are 4 tweets in Figure 4.3. The first 3 of them are about the makeup data breach and the last one is a general idea about the data breaches posted at same time. Firstly, the collected tweets need to be classified about if they report a cyber security event or not. The expected classification result can be seen in Figure 4.4 below.

After the classification, NER applied to the positive tweets to extract victim information which is 'Sabanci University' in this case. Demonstration of the NER can be seen in Figure 4.5 below.

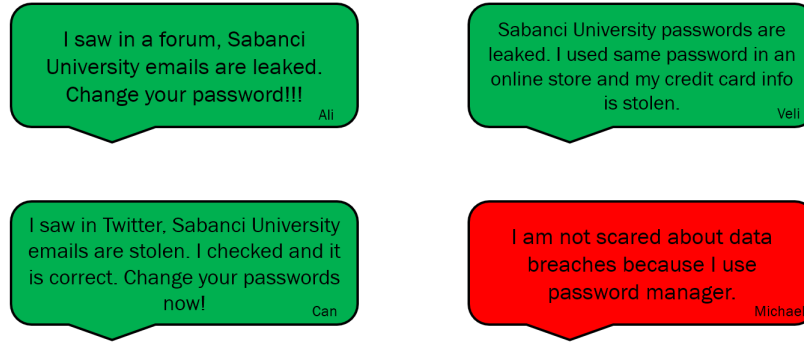


Figure 4.4 Expected Classification Result for Makeup Tweets

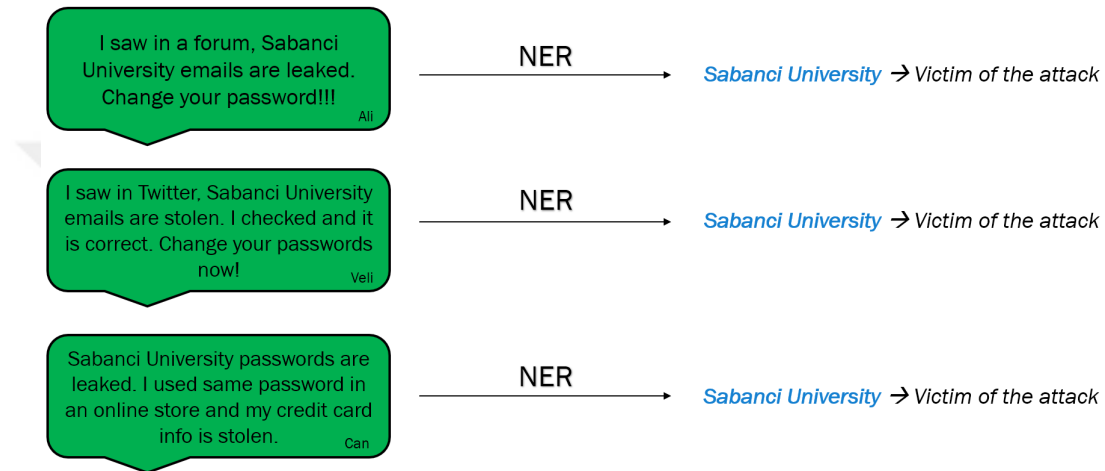


Figure 4.5 Expected NER Result for Makeup Tweets

In this way, members of Sabanci University can be warned, and the IT department of the university can be notified if it is not discovered yet. Thus, people can protect their e-mails and university accounts. Also, they can protect accounts that have the same credentials with university account on another website such as e-commerce website that includes saved credit card information.

This example is for demonstrating our motivation of this thesis. Unfortunately, there are lots of real data breach cases that were hidden from the customers for years although it is crime according to law. Systems as in this study can help people to minimize their loss and learn the real situation which is a legal right of them. Tweets to inform people or to express their opinion can help people in this way and other potential victims can secure themselves even they were not informed by companies.



## 5. Experimental Settings

In this work, different deep learning models and NLP techniques are evaluated by performing several experiments to find the best approaches for the tasks that are explained in Chapter 4. The details of the experimental settings are explained in this chapter.

### 5.1 Classification Datasets

There are two datasets that are used to evaluate the classification in this study. The contexts of the datasets are shared in detail for both of them in this section. Also, the details of the tweet collection and pre-processing that are applied to tweets are explained for our dataset.

#### 5.1.1 SUCyber

A dataset was created and named SUCyber to evaluate the proposed approaches in this work. It is also called as first dataset and our dataset alternatively in this paper. DDoS attack selected as example cyber security event for the dataset creation. For the collection of positive samples, real attacks against 44 different entities on 38 different dates that are between 2010 and 2020 were determined. English tweets posted on attack date about an entity that is under attack were collected by searching tweets that include both '**DDoS**' and '**name of the entity**' keywords as in (Ritter et al., 2015) using Twitter Search API. In this way, it is aimed to collect as many positive tweets as possible. Afterwards, the collected tweets are manually labeled about whether it is related with a cyber event or not. For example, a DDoS attack

was carried out against Wikipedia on 6 September 2019. The search query for the example attack is as follows:

*"Wikipedia DDoS lang:en until:2019-09-07 since:2019-09-06"*

This query returns us English tweets that include both keywords that are 'Wikipedia' and 'DDoS' posted on September 6, 2019. The full list of queries that were used to create the datasets of this work are available in the Appendix A to examine all selected attacks and their corresponding dates. The information can also be used to reproduce the dataset with the available tweets.

More than 15,000 tweets were collected. Collected tweets were manually eliminated to keep tweets as unique as possible and they were also manually labeled as positive or negative. If a tweet is about a DDoS attack, it was labelled as positive, if not it was labelled as negative. However, there were very similar tweets among the collected tweets. For example, if a famous cyber security related person posts a tweet about an attack, it is re-shared by lots of people sometimes by adding something to it, sometimes not. Also, there were tweets that are very similar to each other such as the only difference between them is that only a letter or word. In order to have unique tweets in the dataset, tweets that are more than 80% similar to each other in terms of their longest common subsequences are discarded if they do not have meaningful additional information.

As a result of the tweet selection and labelling process, 4013 positive tweets for DDoS attack were obtained. In this collection, a few numbers of negative tweets were collected due to the tweet collection approach. Therefore, we collected 2019 negative tweets for the DDoS attack separately. The negative tweets are related to DDoS attacks but, they do not report an attack i.e. they are negative in this context. The collected negative tweets are about the following aspects:

- General information related to DDoS attacks
- Advertisement of products for DDoS attack protection
- Seminars and strategies about DDoS attacks
- DDoS attacks detection techniques
- Statistical news about DDoS attacks

**Pre-processing** Pre-processing is applied to the collected tweets to prepare them to use in deep learning models. Firstly, special characters are removed. Then, URLs, emojis and reserved words (RT and FAV) are discarded. Lastly, punctuations are erased, and all characters are transformed to lower-case. An example result of the process can be seen Figure 5.1 below.

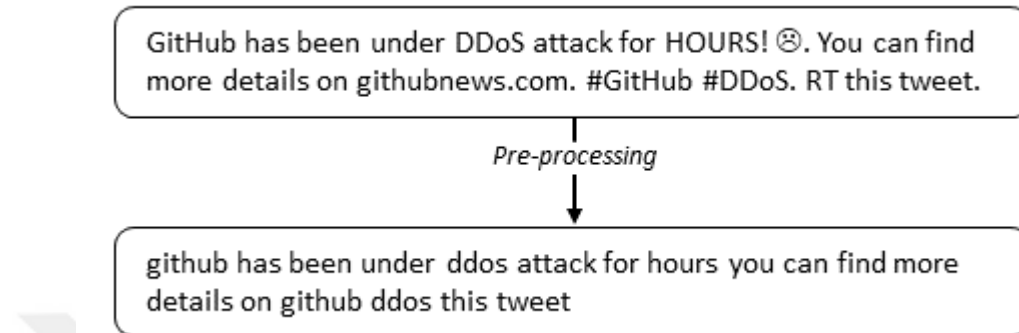


Figure 5.1 Example for Tweet Pre-Processing

### 5.1.2 Second Dataset

In another study (Yagcioglu et al., 2019), a dataset was constituted by researchers. The dataset includes 2000 tweets such that 843 are positives and 1174 are negatives. For the collection of the tweets, 6 different keywords were used: ‘**denial of service**’, ‘**botnet**’, ‘**malware**’, ‘**vulnerability**’, ‘**phishing**’ and ‘**data breach**’ to collect tweets about these types of cyber events between 2015 and 2018. The dataset that includes the pre-processed tweets is public, so we also decided to use it for the evaluations of the models.

## 5.2 Named Entity Recognition Dataset (SUCyberNER)

For the NER evaluation, we manually annotated 1474 randomly selected positive tweets from our dataset, so a dataset for the NER task was created and named SuCyberNER. The main aim is to extract information of victim of an attack because it is the most important information that we want to achieve. In addition, it was possible to create a dataset for this purpose because all tweets in our dataset have information of the victim. According to the victim information, the entity can be aware of the attack and can start trying to eliminate the attack as fast as possible and protect itself. Also, users of the entity, which is under cyber attack, can learn the situation and act accordingly for their own protection.

In addition to the victim information, there can be other details of an attack that can be extracted from tweets such as attacker, power of attack, duration of effect of attack and specific reason of attack if it exists. However, these details are either learned later or never revealed. The aim of this work is detecting an event as fast as possible so the dataset was constituted according to this aim. Therefore, there are a few numbers of tweets that include this information in our dataset, so it is not meaningful to evaluate these entities with this dataset. There is a need for dataset expansion to create samples for different entities if the list of entities is wanted to be diversified.

Finally, our NER dataset has 1474 annotated tweets, and the only label is '**Victim**' to get information of victim of the attack. Also, other words that do not have useful information are annotated as '**O**'. Lastly, there are 2887 tokens annotated as '**Victim**' and there are 25174 tokens annotated as '**O**'. In addition, the tweets are separated randomly as 1000, 237 and 237 for train, validation and test set respectively.

## 5.3 Baseline for Classification Task

For the binary text classification, there are very specific and fine-tuned systems in specific domains. However, the dataset in this work is limited and relatively small. Also, a more complex model needs more data, and this detail should be considered in the model selection phase. Therefore, more general models are evaluated for baseline

model selection. The same situation for the model selection was encountered in all examined studies as can be seen in Chapter 3. The recent studies use general models such as CNN and Recurrent Neural Network (RNNs), and they achieved good results. For these reasons, CNN and BiLSTM were selected as baseline model candidates for tweet classification.

A CNN model, that is similar to the model of (Dionísio et al., 2019), is the first baseline candidate. The other candidate is a BiLSTM model similar to the model as used in (Fang et al., 2020) for the baseline of text classification task. Unlike (Dionísio et al., 2019) and (Fang et al., 2020) models, the dropout layer before the softmax layer is not added for both models in this paper. Models are evaluated with no dropout layer and 0.5 dropout rate and it worked better without the layer. The possible reason for this outcome is that since the dataset is not large, further regularization may cause a decrease in performance. Details of the models can be seen in below. In this evaluation for the baseline model selection, scores of all alternative approaches are very close to each other and we chose the best approach among them according to small differences.

### 5.3.1 CNN Model

It consists of five layers: input, embedding, convolution, max-pooling, and output. The input layer expects tokenized integer representations of tweets. The next layer is the embedding layer that is used to create an embedding matrix which has rows for each words and columns for values of word vectors. For the word embedding, 3 different alternatives were evaluated: creating randomly and updating during the training, using pre-trained Word2Vec (Mikolov, Chen, Corrado & Dean, 2013) and GloVe models (Pennington, Socher & Manning, 2014). Then, a convolution layer is performed to create feature maps from the embedding matrix created the previous layer according to kernel and filter size. The following layer is the max pooling layer that performs selection over the feature map to reduce it for preventing overfitting and decreasing the computational complexity. Lastly, a fully connected softmax layer is performed to give the probability for the binary tweet classification. Table 5.1 shows the evaluated parameters for the CNN model. The details of the best model is as follows: number of filters is 256 and kernel size is 3.

Table 5.1 Evaluated CNN parameters

Dataset	Kernel	Number of Filters
DDoS	[3, 5, 7]	[128, 192, 256]

### 5.3.2 BiLSTM Model

Similar to the CNN, the same alternatives for word embedding are evaluated with the BiLSTM. The model consists of two LSTM, they move in opposite directions: forward and backward. Although it has a simple structure, it works very well because it obtains better context information thanks to its bidirectional mechanism. For the last layer, a fully connected softmax is located and it results the probabilities for the classes about whether it is related to cyber event or not. For the best BiLSTM model, the dimension of hidden vectors is 64.

### 5.3.3 Word Embeddings

In the models, different word embeddings are evaluated for creating an embedding matrix as briefly stated in model descriptions. The first choice is to initialize word embeddings randomly and update them with backpropagation during training. Vectors sizes of 100, 200 and 300 are evaluated for this approach. Second option is using pre-trained Word2Vec (Mikolov et al., 2013) and GloVE models (Pennington et al., 2014). There are two alternatives within pre-trained embeddings, either the vectors can be kept as it is, called static or the vectors are also update during the training, called dynamic. Also, tweets are usually not written in formal language; therefore, vector representations of some of the words do not exist. In such a case, random vectors were initialized for unknown words for pre-trained models. In addition to these, fastText (Bojanowski, Grave, Joulin & Mikolov, 2017) was evaluated but performs worse than all other alternatives.

## 5.4 Baseline for NER Task

LSTM-CRF model (Lample, Ballesteros, Subramanian, Kawakami & Dyer, 2016a) are one of the most successful approaches for the NER task. In addition, (Dionísio et al., 2019) uses BiLSTM-CRF approach that is based on work done by (Lample et al., 2016a) for the NER task. For these reasons, BiLSTM-CRF approach is selected as a baseline model for the NER task to extract information from tweets that are related to a cyber security event. The results of the baseline model on our dataset were obtained by using publicly available implementation <sup>1</sup> of it.



---

<sup>1</sup>[https://github.com/guillaumegenthial/sequence\\_tagging](https://github.com/guillaumegenthial/sequence_tagging)

## 6. Experiments

In this chapter, evaluation results of all approaches are shared. Precision, recall and F1-score of models are shared and best model is selected according to the F1-scores. The classification results for both datasets are the mean of 10-fold cross-validation experiments. Also, standard deviations of the results are shared. More details of the models can be found in Appendix A to analyze and reproduce the experiments.

### 6.1 Tweet Classification

In order to achieve better results than the existing approaches for the binary classification task, several experiments are made and shared in this section. As mentioned before, we have 2 separate datasets: the first dataset that we created and the second dataset of another study (Yagcioglu et al., 2019), so the results of them are shared separately.

For the task, several experiments are made by using transfer learning and transformers along with the baseline models. Results of ELMO embedding (Peters, Neumann, Iyyer, Gardner, Clark, Lee & Zettlemoyer, 2018), ULMFiT (Howard & Ruder, 2018), BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019), and XLNet (Yang et al., 2020) for binary classification on the datasets are shared.



## 6.2 Classification Results

The classification results on the two datasets are shared in the following sections separately. Also, there is a evaluation on combined dataset at the end of the section. The performances of the all evaluated models demonstrated and explained in detail.

### 6.2.1 Results for First Dataset (SuCyber)

Firstly, Support Vector Machine (SVM) is trained to see the performance of a machine learning model on the test dataset, and it achieves 94.92% F1-score. It is an expected result to reach the lowest result with the SVM but it is useful to compare deep learning approaches with the machine learning approach for the task to see the general difference. For the baseline models, CNN with GloVE embeddings performs best in all evaluated CNN variants, it achieves F1-score of 97.89%. The reason of being best of GloVE among other embeddings is because it was pre-trained on tweets so it is more compatible with our datasets. In addition, the other baseline model that is BiLSTM by using randomly created word embeddings that have a size of 200 achieves the best performance among the all evaluated BiLSTM approaches, the F1-score for the model is 97.34%.

The ELMo embedding approach decreased the performance by resulting F1-score of 95.08% because of the structure of tweets, which are usually informal short texts, and the sizes of the datasets. The datasets are relatively small so creating different word embeddings according to context did not work as expected on small number of the short unstructured texts. After that, ULMFit is evaluated for the binary classification and it provides to overcome the baseline results with F1-score of 98.10%. It is an expected result because ULMFit is still successful for text classification and our datasets are relatively small so the advantage of transfer learning provides this increase in the performance. We fine-tuned the pre-trained language model and classifier on the datasets. Therefore, it is expected to work better in our datasets because the model was pre-trained on a larger dataset so it is more powerful. Then, BERT, which can perform better than ULMFiT in general for NLP tasks such as sentiment analysis, is evaluated to increase the performance further. As expected, it yields F1-score of 98.69%. After this point, RoBERTa was selected to evaluate the task on our dataset because it was created to increase the performance of BERT.

Unsurprisingly, it provided to achieve the best results with F1-score of 98.89%. Alternatively, XLNet was evaluated because it keeps state-of-the-arts results for text classification on some benchmarks. However, it performs worse than both BERT and RoBERTa. Both precision and recall slightly decreased with the XLNet.

More detailed discussion about the results can be found in the next chapter which is chapter 7 with more insight of the models. Consequently, the best performance is F1-score of 98.89% and it is achieved by using RoBERTa for the binary classification on the SUCyber. All results of the first dataset can be seen in Table 6.1.

Table 6.1 Classification Results for SUCyber

Model	Precision		Recall		F1-score	
	Mean (%)	SD	Mean (%)	SD	Mean (%)	SD
SVM	94.90	.0094	94.96	.0139	94.92	.0077
CNN-Glove	96.84	.0119	98.97	.0037	97.89	.0056
BiLSTM-Random	96.44	.0108	98.28	.0070	97.34	.0055
ELMo	96.29	.0083	93.96	.0267	95.08	.0118
ULMFiT	97.20	.0113	99.02	.0083	98.10	.0035
BERT	97.97	.0080	99.42	.0066	98.69	.0021
RoBERTa	98.28	.0051	99.52	.0047	<b>98.89</b>	.0029
XLNet	97.77	.0086	99.20	.0055	98.47	.0029

### 6.2.2 Results for Second Dataset

The second dataset which was created in the different study (Yagcioglu et al., 2019) are also used to evaluate the approaches in this work. In order to achieve fully comparable results, we contacted the authors, but we could not able to get exactly the same data split. Therefore, we implemented their approach with all the details as possible, so we could be able to use same data splits and we got comparable result with our study.

The SVM yields the worst performance among all as expected. The approach in (Yagcioglu et al., 2019) gave mean F1-score of 72.02% according to our implementation on a randomly selected data split. We evaluated our baseline models on the same data split, the F1-score reached to 75.66% with CNN-GloVE that is also the best baseline model for the first dataset. After this point, ELMo embedding is evaluated but it cannot overcome the baseline result like in the SUCyber dataset, it gives F1-score of 75.63%. The performance is slightly decreased as in the first dataset because of the same reasons which are size of datasets and writing structure of tweets. The situation is more obvious in the second dataset because the dataset is smaller than the first dataset. Also, positive samples are less than negative samples. Thus, the model cannot be trained with a sufficient positive samples in comparison with the SUCyber. As a result, it gives more false negatives in the less positive sample. Therefore, there is a need for more training data to capture the context better. However, ULMFiT results F1-score of 77.19% for the task and it makes the model the best one evaluated so far. Although this dataset is even smaller than the SUCyber, the model still worked as expected thanks to the advantage of transfer learning. In addition, BERT and RoBERTa are evaluated on the dataset to get better results as in SUCyber. The F1-scores are 80.40% and 81.30% respectively. Their precision results are very close but RoBERTa yields better recall score. It means that RoBERTa predicts the actual positives more accurately for more tweets. Lastly, XLNet is evaluated on the second dataset but it performs worse than BERT and RoBERTa in terms of F1-scores. Although it yields the highest recall score, it gives the worst precision among the 3 models. The balance between precision and recall is also important and this situation is also reflected in the F1-Score of XLNet.

More evaluation about the results can be found in next chapter with more insights of the models. Consequently, the best results for the binary classification on the second dataset is F1-score of 81.30% and it is achieved by RoBERTa. All results of the second dataset can be seen in Table 6.2.

Table 6.2 Classification Results for Second Dataset

Model	Precision		Recall		F1-score	
	Mean (%)	SD	Mean (%)	SD	Mean (%)	SD
SVM	75.71	.0496	68.47	.0467	71.84	.0424
CNN-LSTM [1]	73.58	.0384	71.07	.0652	72.02	.0270
CNN-Glove	76.61	.0350	74.89	.0302	75.66	.0210
BiLSTM-Random	74.75	.0343	73.89	.0923	73.97	.0488
ELMo	80.28	.0592	73.15	.1107	75.63	.0377
ULMFiT	80.16	.0446	74.78	.0803	77.19	.0568
BERT	78.40	.0497	83.17	.0694	80.40	.0343
RoBERTa	77.82	.0597	85.89	.0631	<b>81.30</b>	.0276
XLNet	74.48	.0546	88.00	.0774	80.25	.0325

### 6.2.3 Results for Combined Dataset

In this subsection, we aimed to perform cross check for the evaluated approaches. For this purpose, we used our dataset for training the models and used the second dataset for testing. The challenges and the results of this evaluation are given below.

First of all, our dataset consists of 6092 tweets for only DDoS attacks. On the other hand, the second dataset consists of 2000 tweets for 6 different cyber event types as ‘denial of service’, ‘botnet’, ‘malware’, ‘vulnerability’, ‘phishing’ and ‘data breach’. Other details such as the methodology of collecting and labeling tweets about the second dataset are not shared. The only information shared is the total number of tweets. Further, the number of tweets for these events are not shared.

The other factor which makes harder this evaluation is the approach of labeling. The detail of labeling is not shared in the second dataset but when we examined it we found that there are some tweets about DDoS attack labeled positive by them and they should be negative in our dataset. The difference comes from the difference between the ultimate aim of the studies. In their work (Yagcioglu et al., 2019), they only aimed to detect whether a tweet is related with a cyber event or not. Therefore,

the tweets which include general information or statistics about DDoS attack are positive in their dataset. However, our aim is to detect cyber attacks as fast as possible so the same tweets should be negative in our dataset because they do not report a cyber attack. For example, the tweet below was labeled as positive in their dataset but it should be negative according to our labeling approach.

***"report finds 13 increase in ddos attacks on health care since 2016"***

The other example below which is labeled as positive but it should be negative in our approach as explained in experimental settings chapter because it is a general information about DDoS attack and it does not report an attack.

***"ddos real threat that big data can help combat computerworld"***

The last example is the tweet below and the same disparity exists in this case.

***"Kaspersky: DDoS attacks growing stronger with unsecured IoT"***

Moreover, there is an example tweet below which is labeled as negative from them but it should be positive for bot datasets. However, the reason for this inconsistency could not be understood.

***"site is down our dns provider is currently being ddos attacked hope to restore service soon"***

Our detailed analyses show that most of the tweets that are related with DDoS are like the examples above. It means that only a small portion of the tweets classified as positive in the second dataset are positives according to our labeling methodology. Also, there is inconsistency in the labeling as in the last example above. For these reason, the number of common samples for the DDoS attack, which is already small, decreased even more.

Due to the reasons explained above, it is irrelevant to evaluate our approaches by training on our dataset and testing on the second dataset directly. Instead, we created a new test dataset as a subset of the second dataset. We randomly selected 200 tweets and update their labels if necessary according to our labeling approach. As a result, we ended up only 15 positive tweets and 185 negative tweets. For the transformers (BERT, RoBERTa and XLNet) that outperform all other approaches evaluated, the results are as follows.

Firstly, we have only 15 positive tweets in the test set so the results are highly sensitive when we share them as percentage. When we look at the results, the number of false positive is too many, so we reached low precision results. The first reason of the low precision is the few number of positive tweets in the test set. The

Table 6.3 Combined Dataset Results

Model	Precision		Recall		F1-score	
	Mean (%)	SD	Mean (%)	SD	Mean (%)	SD
BERT	49.05	.0901	91.11	.0314	63.20	.0784
RoBERTa	31.37	.0744	93.33	.0544	46.19	.0765
XLNet	27.67	.0563	84.44	.0628	41.28	.0601

other reason is most of the negative tweets in the test set are too different than the tweets in our dataset because of the difference of the event types. On the other hand, we reached 93.33 % recall with RoBERTa because our models are trained on our dataset and the positive tweets in the test set are compatible with SUCyber. It means that the models yield few number of false negatives. The tweet below is the example of false negative prediction.

*"new mirai variant unleashes 54hour ddos attack networksecurity newsampindustry"*

At first sight, it looks like the tweet reports a DDoS attack which continues for 54 hours so it was labeled as positive. However, it can also be deduced that this is a statistical news about a new Mirai botnet variant. Therefore, it is an acceptable mistake for the model.

### 6.3 Named Entity Recognition and Results

The NER results are obtained by taking the mean of 3 different run of the same experiment. The evaluated approaches are CRF, BiLSTM-CRF (Lample, Ballesteros, Subramanian, Kawakami & Dyer, 2016b), BERT, RoBERTa, and XLNet. Among all of the models, BERT gave the best result for the task. It is also used in many NER tasks such as (Tikhomirov et al., 2020) used it for cyber security domain in Russian texts. Therefore, it is selected to use for the NER task for the proposed framework. It was implemented by using the pre-trained model and it was fine-tuned on SUCyberNER dataset.

CRF model gives F1-score of 82.18% for the Victim entity. The baseline model which is BiLSTM together with CRF (Dionísio et al., 2019) results F1-score of 86.85% for the entity. The combination of the models increased the performance because BiLSTM provides to capture the better sequential relationship of tokens so it increases the overall performance. Moreover, we evaluated BERT to get a better result and we achieve F1-score of 92.29% for the entity of 'Victim'. In addition, XLNet gives F1-score of 91.87% for the same tag. Lastly, RoBERTa was evaluated and it gives F1-score of 91.88%. BERT and XLNet provide more balanced performance in terms of precision and recall. However, RoBERTa yields the best recall score among all but the precision is lower. It means that the number of false positive is higher in RoBERTa than BERT and XLNet. As a result, the BERT approach gives better scores than the CRF, BiLSTM-CRF, RoBERTa, and XLNet approaches for the victim entity in our dataset for NER task. All results of the models can be seen on Table 6.4.

Table 6.4 NER Results for Victim Tag

Model	Precision (%)	Recall (%)	F1-score (%)
CRF	86.94	77.92	82.18
BiLSTM-CRF (Dionísio et al., 2019)	89.16	84.66	86.85
BERT	92.18	92.42	<b>92.29</b>
RoBERTa	89.39	94.55	91.88
XLNet	91.77	91.97	91.87

## 7. Discussion

The discussions about the results of experiments are shared in this chapter. In other words, the reasons why a model performs better or worse than the others or the strengths and the weaknesses of the models are reviewed.

### 7.1 Tweet Classification

In this work, binary classification results for the 2 different datasets is shared as can be seen in Chapter 6. Both dataset consists of tweets about cyber security events. However, SUCyber consists of tweets about only one attack type but the other dataset includes 6 different event types. Also, our dataset consists of 6092 tweets, but the second dataset includes 2000 tweets. As can be observed, the results of SUCyber dataset is at around 98% and the results are at around 80% for the second dataset so there is a general difference between them. The reason of the difference between scores on the two datasets is based on the size and the context of the datasets as mentioned above. Briefly, the models that we evaluated can perform better by training them using more datasets in general. In the first dataset, we have more samples about a cyber event so the model can learn better. Therefore, the scores for the first dataset are higher than the second dataset in this case.

There is a similar trend that can be noticed in both results for the classification. The ELMo embedding decreases the performance slightly when we compare it to both baseline models on the first database and this is the same if we compare the CNN baseline with ELMo in the second dataset. In addition, ULMFiT increases the performance of the classification and overcomes the performance of ELMo. Moreover, BERT, RoBERTa, and XLNet perform better than the other alternatives. BERT performs better than XLNet but RoBERTa is the best among all of the 3 models. These datasets consist of tweets about different cyber events with different numbers



of samples and it is good to observe the trend on both. This situation shows us the approaches are not dependent on the tweets contained in the datasets or the event types of tweets. Therefore, any new event type can be added to the system to detect it from tweets.

As we generally expected in the evaluations, either we surpassed the previous approach or got a result close to it in the worst case. However, this was different for the ELMo embeddings because we expected an increase with it at the first sight. The model provides to have different word embeddings for the same word according to its context different from the traditional embeddings and this feature provides an increase in performance for some cases. However, ELMo embeddings decrease the performance as can be seen in Chapter 6 for this case. It gives lower recall scores which means that the Type II error rate of the model is too high. The model predicts positive classes as negative falsely. The situation is more obvious in the second dataset because the dataset is smaller than the first dataset. Also, positive samples are less than negative samples. Thus, the model cannot be trained with a sufficient positive samples. As a result, it gives more false negatives in the less positive sample. The reason for this problem is that there is a need for more training data to capture the context better. Also, tweets usually are not written in formal language, and they are short text. ELMo can work well with longer and more organized sentences or with bigger datasets. However, for the tweets in the relatively small datasets, using pre-trained word embeddings especially GloVe (Pennington et al., 2014) that was pre-trained on tweets can perform better as in our evaluations for CNN baseline model.

ULMFiT is one of the most successful transfer learning methods in NLP and it gives good results for text classification in general. Our datasets are relatively small, so transfer learning is very useful for this case to get better results. Although it is advantageous, it is surprising that it was not adequately evaluated in current studies for the cyber security domain. We fine-tuned the pre-trained language model and classifier on the datasets so that it is expected to work better in these datasets because the model was trained on a larger dataset and it is more powerful. As expected, ULMFiT performs better than the SVM, CNN, BiLSTM, and ELMo approach.

BERT is a breakthrough in NLP for many tasks. The reason for the success is based on its design that includes bidirectional language representation and the huge size of the dataset that was used for training. Therefore, it is also one of the best approaches for also the classification. Also, XLNet is very successful in text classification. It has 2 different variants as base and large. In this work, base version that was based

on same data of BERT is evaluated. The difference between BERT and XLNet is mainly based on language modeling. In XLNet, all tokens are predicted in random order but only 15% of tokens, that are masked, are predicted in BERT. At this point, we can observe that XLNet decreases the precision which means that rate of false positive increased. A possible reason of this decrease is the form of tweets that are generally short texts. We can observe that datasets consist of longer text cases where XLNet provides to best result. In short texts, language modeling of the BERT model can be more successful as our case. In addition to these models, RoBERTa was trained with 160 GBs more data than BERT by introducing a new masking methodology. Thus, RoBERTa aimed to increase in performance in some NLP tasks. It yields better precision and recall score than BERT in SUCyber. On the other hand, there is a trade-off between recall and precision in the second dataset. It increases the recall but decreases the precision, and the reason for this is the size of the dataset which is smaller than the first dataset, so the values on the results for the dataset is more sensitive. Also, there are more negative tweets than positive tweets in it. However, RoBERTa also yields the best F1-score among all other approaches in the second dataset as in SUCyber. Consequently, RoBERTa gives the best results among all approaches in both datasets as expected because it is one of the most successful deep learning models that we can use for classification nowadays.

**Analysis of Training Data Size** The effect of the training data size to the results was analyzed. The aim is to observe minimum or optimal size of the training data for a cyber attack type. Since our dataset consists of tweets which are only about DDoS attack, we can observe the changes on the performance when we change the size of the training data for an attack type. Therefore, this information can be useful for the future experiments when a new cyber attack type is wanted to be added to the evaluation. For this purpose, we evaluated the best model, which is RoBERTa, by using smaller samples for the training data starting from the 4748 tweets on the same test set. The results of the model on decreasing training data size can be seen on Table 7.1.

In general, more training data provides better performance. However, the situation can differ in different data samples. Also, it is difficult to give direct number for optimal training data size but if we observe the results, the performance is decreased sharply between 1248 and 748. Therefore, 1000 tweets can be a critical threshold. Also, the powerful models yield F1-Score of 98.47% or more when we check the all result of SUCyber on Table 6.1. Therefore, size of the minimum training set can be 2500 to stay above F1-Score of 98% and the optimal size can be around 3000

Table 7.1 Effect of the Training Data Size

Training Data Size	F1-Score (%)
4748	98.89
4248	98.57
3748	98.35
3248	98.21
2748	98.01
2248	97.87
1748	97.41
1248	97.38
748	95.16
248	93.61

or more for better results. However, more data can provide better performance in general as mentioned above. Therefore, these sizes that we discussed can be used to determine the beginning point for creating data samples to add new cyber attack types.

## 7.2 Named Entity Recognition

For the NER task, BERT performs better than the BiLSTM-CRF (Dionísio et al., 2019) approach which is state-of-the-art in some benchmarks. BERT was also used for NER tasks in the cyber security domain especially for Russian by Tikhomirov et al. (Tikhomirov et al., 2020). Therefore, it is not a surprising result to see this improvement with BERT for the NER task for the entities. The most important entity that we want to achieve from the positive tweet is '**Victim**' and the F1-score of 92.29% is achieved. It was 86.85% with the BiLSTM-CRF (Dionísio et al., 2019) approach.

Consequently, transfer learning and transformers are one of the most successful approaches in NLP nowadays. Also, the datasets that we have are small so, we utilized the power and advantage of transfer learning and transformers. In this way, better results for the tasks are obtained without training models from scratch with small datasets that are used in this work.

Since the result of the NER model is given as the mean of 3 different run, we randomly select a run to analyze NER performance on the test dataset. The aim here is that observe if there is a repeating fault or specific problem in the result.

In our observations, there is only one case the attacker information predicted as a victim for a tweet on the test dataset about a DDoS attack against Twitch from a hacker group. In addition, there is also a remarkable result about the following tweet that shares the DDoS attack against the list of victims.

***"tpyle76. (2016, October 21). Here is a list of sites affected by the DDoS attack today. ActBlue Basecamp Big cartel Box Business Insider CNN... [Tweet].***

***<https://twitter.com/tpyle76/status/789539804002197504>***

The DDoS attack generally targets a victim but the attack can cause emerging other victims. Mostly, a user does not notice all of the victims of a cyber attack because possibly it is not the user of all of them. Even if users notice all of them, they usually only share about the most popular or the most used one. However, fewer users share tweets that is similar to the example above. There are 6 victims in the example tweet above as Act Blue, Basecamp, Big Cartel, Box, Business Insider, and CNN. However, our model only predicted correctly ActBlue that is the first reported as a victim. It is because most of the tweets in the dataset consists of tweets that include few number of victims, mostly one victim. Therefore, the model can perform worse in such tweets.

There are also some cases such as victims predicted as not a victim in results. The common feature in such cases is they are either abbreviations of a long name. For example, WoW (World of Warcraft), OW (Overwatch), COD (Call of Duty) and, XBL (XBOX Live). However, there are also more than one victim name in most of these tweets so there may not be a direct connection with abbreviations. The actual reason of the fault can be the observation made above for tweets that include multiple victims. There is an example tweet below from the test set. PSN is successfully predicted as a victim but XBL is not.

***"spydervenom. (2014, December 26). Here is a list of sites affected by the DDoS attack today. So I just heard about the XBL amd PSN DDoS attacks. Looks like I'll be playing retro tonight. [Tweet].***

***<https://twitter.com/spydervenom/status/548249145257910274>***

The last example is a tweet shared below from the test dataset and it is predicted incorrectly by the NER model. The tweet includes irony in terms of meaning. It is about a DDoS attack against BBC News but the author of the tweet used the name for both victim and attacker so it can be expected that the victim information cannot be predicted accurately.

***"BBCPropaganda. (2015, December 31). BBC launches DDOS attacks on TV file sharing sites = not news, just business Anonymous launches DDOS attacks on the BBC = end of the world [Tweet].  
<https://twitter.com/BBCPropaganda/status/682524801672581120>"***

Apart from the analyses above, specific and recurrent errors were not noticed in the results. This means that the performance of our NER model can be improved further by using more tweets for training.

## 8. Website

We created a website <sup>1</sup> to show the performance of the proposed approach in real-time for DDoS attacks. The proposed approach is implemented and the results can be examined on the website. Filtered stream API of Twitter is used but all posted tweets cannot be collected because of the limitation of the API. However, all successfully collected tweets are processed by our framework. As mentioned above, tweets are grouped by victim information day by day. Example grouping can be seen in Figure 8.1. On the website, users can select a date to observe tweets about victims if any. Because of the Twitter privacy policy, we can only share the ID of a tweet so the IDs listed as clickable links on the website. If a tweet still exists and it is publicly available user can view it on Twitter. In conclusion, IDs of all positive tweets classified by our approach and the number of tweets posted for victims can be observed for each available day on the website.

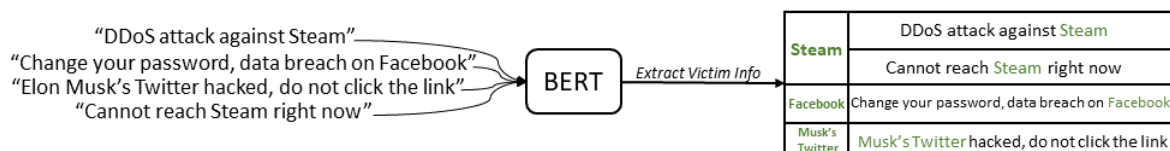


Figure 8.1 Example of Named Entity Recognition and Grouping Tweets

An entity can be named in different ways, for example "**Solana**", "**Solana Blockchain**" and "**Solana Network**". Actually, all of them represent the same entity. In this type of situations, the tweets that belong to the same victim, which is called in different ways, can be combined. Firstly, we find the victims that can be combined by checking common words as in "**Solana**" example. In a day, if two or more victims have common words in them, they probably target to the same entity in different ways. However, we also added an extra precaution to prevent incorrect combination of victims. A similarity measurement according to the semantic of the tweets is applied by using Sentence-BERT (Reimers & Gurevych, 2019) model for the tweets of the victims that are candidates for combination.

<sup>1</sup>sutect.com

Among all possible victims for the combination, we find the victim with most tweets about it. For the rest of the victims, we measure semantic similarity between tweets of them and tweets of the most crowded victim that we found. If the similarity result is above 0.5, the victims are combined under the most crowded victim. For instance, suppose there are 250 tweets for **"Solana"**, 50 tweets for **"Solana Blockchain"** and 30 tweets for **"Solana Network"** in a day. We select **"Solana"**, which is the most crowded victim, among all. The similarity between the tweets of **"Solana"** and tweets of **"Solana Blockchain"** are calculated. If the result is higher than 0.5 they are combined under the **"Solana"** victim. The same process is performed for the **"Solana Network"**. At the end of the process, these victims are combined under **"Solana"** victim.

**Example Result from Website** Named entity that is extracted from our NER model, the number of tweets related with that entity, and IDs of these tweets can be examined on the website day by day. There were DDoS attacks on different scales in September 2021. For example, Yandex that is a Russian web service experienced a powerful DDoS attack. As an example date, 10 September 2021 is selected to view detected DDoS attacks. As mentioned, Yandex is detected from our framework with a total of 98 tweets on this day. There are also other results such as KrebsOnSecurity and ANZ New Zealand in the result table. After the confirmation by researching, we found that DDoS attacks also occurred on KrebsOnSecurity website which is a news website about cyber security together with Cloudflare. Also, a news website called ANZ New Zealand was experienced a DDoS attack. All of these attacks were executed by using a botnet called Meris. In the below, a detailed screenshot from the website for that day can be seen in Figure 8.2. In addition, the threshold is set as 7 for the tweets about a victim for this demonstration.

The website gives all the tweet IDs as a link in a list when a user clicks **"See Tweet IDs" button**. This step can be seen in Figure 8.3.

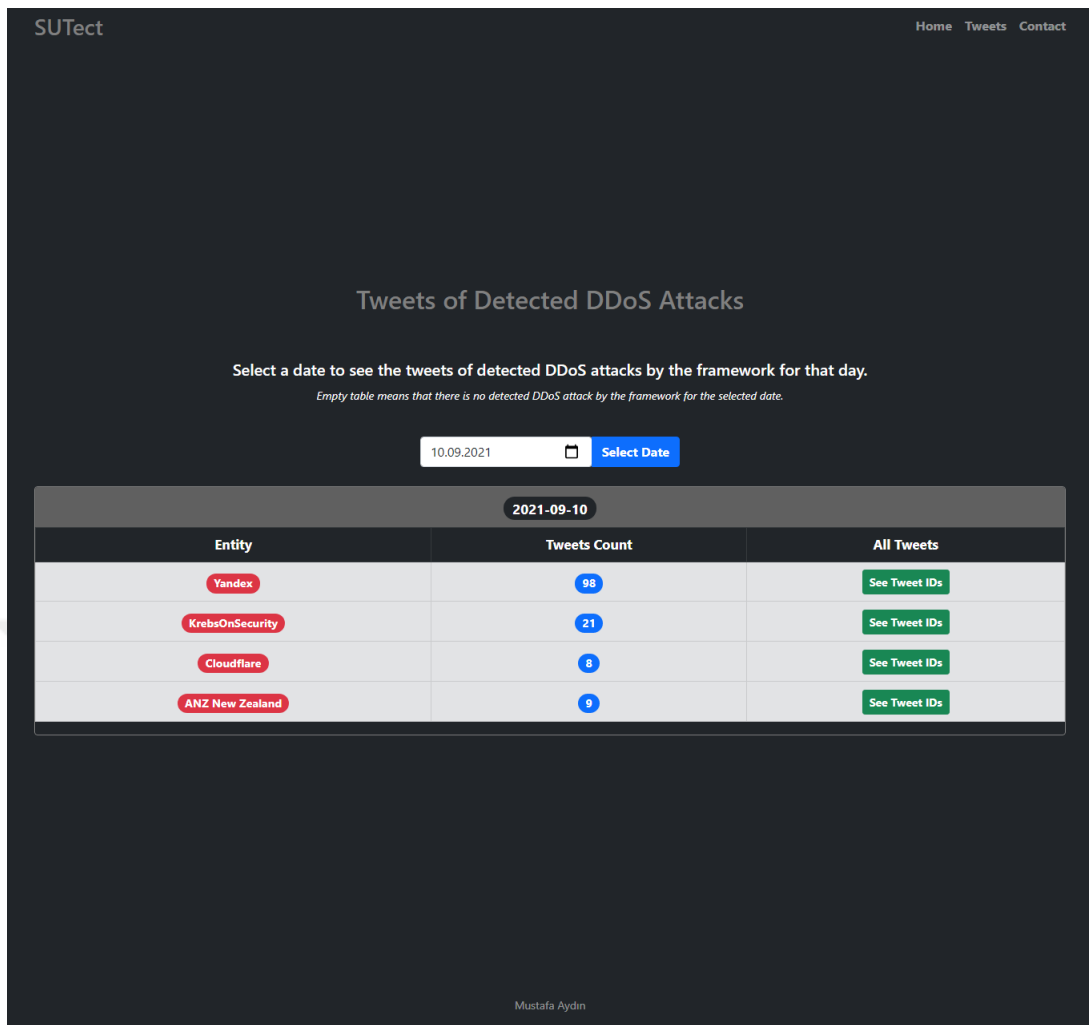


Figure 8.2 Result table from the website for September 10, 2021.

After clicking on a randomly selected ID from the list, we observed an example tweet about Yandex and it can be seen in Figure 8.4.



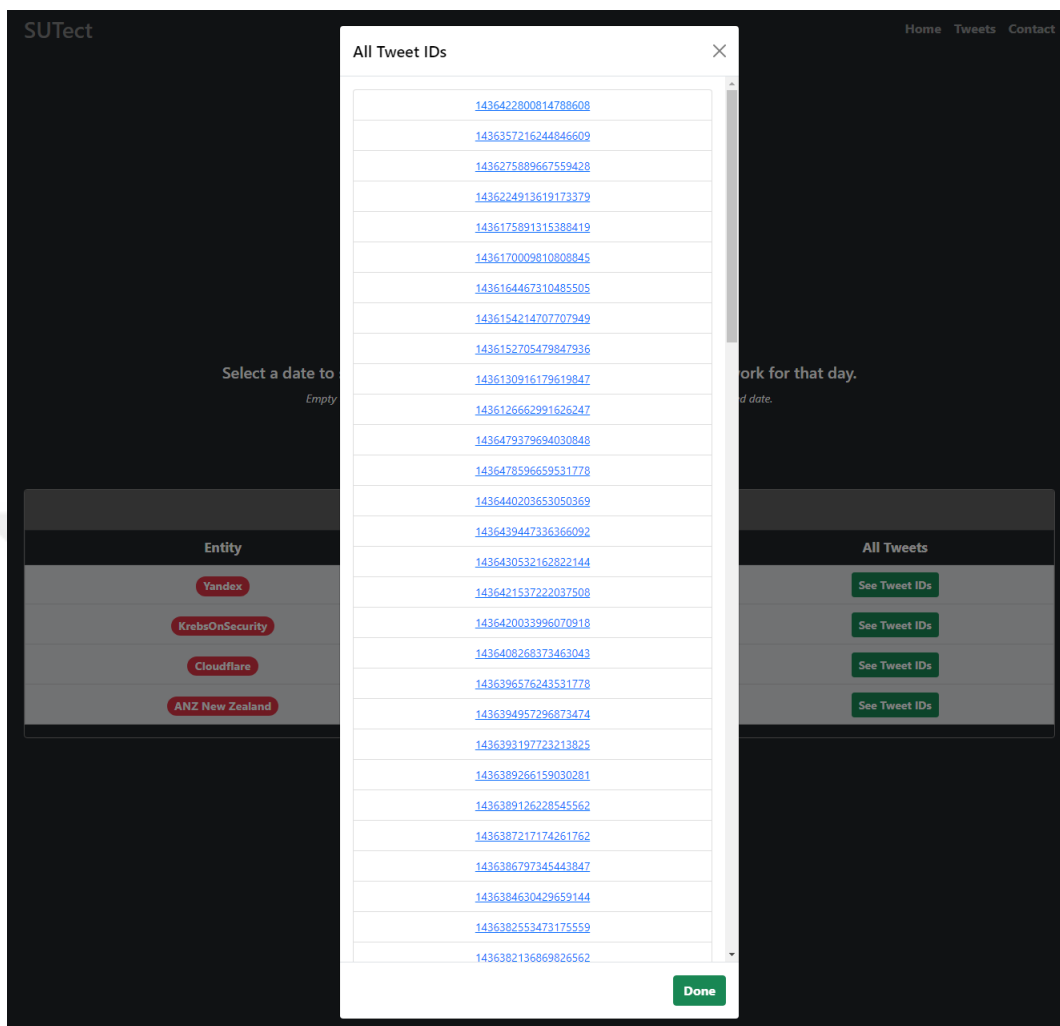


Figure 8.3 Example list of tweet IDs for Yandex from the website for September 10, 2021.

Russian internet giant #Yandex has been targeted in a massive distributed denial-of-service (DDoS) attack that started last week and reportedly continues this week.  
[#DDOS](#) [#CyberSecurity](#) [#CyberAttack](#) [#RuNet](#) [#Russia](#)

Figure 8.4 Example tweet selected randomly from the IDs list for Yandex from the website for September 10, 2021.

## 9. Conclusion and Future Work

This thesis proposes a framework that includes two main sub-tasks: binary classification and NER together with other modules to detect cyber attacks from Twitter as fast as possible. Firstly, tweets are pre-processed, then the classification is performed to determine whether a tweet reports a cyber attack or not. If the tweet is classified as relevant, it is forwarded to the NER to extract information of the victim of a cyber attack. Lastly, it is checked whether the account that sent the tweet is a bot account or not to keep the system as consistent as possible.

A dataset named SUCyber, which consists of DDoS attack related tweets, was created to evaluate approaches in this work. Also, another dataset (Yagcioglu et al., 2019) that includes 6 different cyber events is used to evaluate the approaches. In addition to these, a NER dataset called SUCyberNER was created by manually annotating 1474 random tweets from SUCyber. For the classification task, transfer learning methods and transformers are utilized. The best F1-score is obtained with RoBERTa and it is 98.89% for the first dataset. Also, the best F1-score is 81.30% and it is obtained with RoBERTa for the second dataset. It is 72.02% with the approach of the study (Yagcioglu et al., 2019) that is the owner of the dataset. In addition to these, we aim to extract victim information from the tweets and the F1-score obtained as 92.29% for the victim tag. The score is 86.85% with the BiLSTM-CRF (Dionísio et al., 2019) approach.

Consequently, this work shows that whether a tweet is related with a cyber attack or not can be detected with high accuracy. In addition, information extraction from the relevant tweets about the victim of a cyber attack can be extracted with high success rate. Therefore, Twitter can be used as an information source to detect cyber attacks.

As a future work, Named Entity Recognition can be improved to perform better in terms of precision and recall. Therefore, SUCyberNER dataset is needed to be expanded by adding more tweets. Also, the website that we created for presenting the proposed approach can be enriched with more details and statistical information.

## BIBLIOGRAPHY

- Antonakaki, D., Fragopoulou, P., & Ioannidis, S. (2021). A survey of twitter research: Data model, graph structure, sentiment analysis and attacks. *Expert Systems with Applications*, *164*, 114006.
- Behzadan, V., Aguirre, C., Bose, A., & Hsu, W. (2018). Corpus and deep learning classifier for collection of cyber threat indicators in twitter stream. In *2018 IEEE International Conference on Big Data (Big Data)*, (pp. 5002–5007).
- Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2017). Enriching word vectors with subword information.
- Chambers, N., Fry, B., & McMasters, J. (2018). Detecting denial-of-service attacks from social media text: Applying NLP to computer security. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, (pp. 1626–1635)., New Orleans, Louisiana. Association for Computational Linguistics.
- Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding.
- Dionísio, N., Alves, F., Ferreira, P. M., & Bessani, A. (2019). Cyberthreat detection from twitter using deep neural networks. *CoRR*, *abs/1904.01127*.
- Dionísio, N., Alves, F., Ferreira, P. M., & Bessani, A. (2020). Towards end-to-end cyberthreat detection from twitter using multi-task learning. In *2020 International Joint Conference on Neural Networks (IJCNN)*, (pp. 1–8).
- Fang, Y., Gao, J., Liu, Z., & Huang, C. (2020). Detecting cyber threat event from twitter using idcnn and bilstm. *Applied Sciences*, *10*, 5922.
- Habdank, M., Rodehutsors, N., & Koch, R. (2017). Relevancy assessment of tweets using supervised learning techniques: Mining emergency related tweets for automated relevancy classification. In *2017 4th International Conference on Information and Communication Technologies for Disaster Management (ICT-DM)*, (pp. 1–8).
- Howard, J. & Ruder, S. (2018). Universal language model fine-tuning for text classification.
- K, S., Balakrishna, P., R, V., & KP, S. (2020). Deep learning approach for enhanced cyber threat indicators in twitter stream.
- K, S., S, S., R, V., & KP, S. (2020). Deep learning approach for intelligent named entity recognition of cyber security.
- Lallie, H. S., Shepherd, L. A., Nurse, J. R., Erola, A., Epiphaniou, G., Maple, C., & Bellekens, X. (2021). Cyber security in the age of covid-19: A timeline and analysis of cyber-crime and cyber-attacks during the pandemic. *Computers & Security*, *105*, 102248.
- Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., & Dyer, C. (2016a). Neural architectures for named entity recognition. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, (pp. 260–270)., San Diego, California. Association for Computational Linguistics.
- Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., & Dyer, C. (2016b).

- Neural architectures for named entity recognition.
- Le Sceller, Q., Karbab, E. B., Debbabi, M., & Iqbal, F. (2017). Sonar: Automatic detection of cyber security events over the twitter stream. In *Proceedings of the 12th International Conference on Availability, Reliability and Security, ARES '17*, New York, NY, USA. Association for Computing Machinery.
- Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., & Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach.
- Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space.
- Naaman, M., Boase, J., & Lai, C.-H. (2010). Is it really about me? message content in social awareness streams. In *Proceedings of the 2010 ACM Conference on Computer Supported Cooperative Work, CSCW '10*, (pp. 189–192)., New York, NY, USA. Association for Computing Machinery.
- Netscout (2021). Cyber security & threat intelligence report. Accessed: 2021/12/20.
- Pennington, J., Socher, R., & Manning, C. (2014). GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, (pp. 1532–1543)., Doha, Qatar. Association for Computational Linguistics.
- Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., & Zettlemoyer, L. (2018). Deep contextualized word representations.
- R., V., Alazab, M., Jolfaei, A., K.P., S., & Poornachandran, P. (2019). Ransomware triage using deep learning: Twitter as a case study. In *2019 Cybersecurity and Cyberforensics Conference (CCC)*, (pp. 67–73).
- Reimers, N. & Gurevych, I. (2019). Sentence-bert: Sentence embeddings using siamese bert-networks.
- Riebe, T., Wirth, T., Bayer, M., Kühn, P., Kaufhold, M.-A., Knauth, V., Guthe, S., & Reuter, C. (2021). Cysecalert: An alert generation system for cyber security events using open source intelligence data. In Gao, D., Li, Q., Guan, X., & Liao, X. (Eds.), *Information and Communications Security*, (pp. 429–446)., Cham. Springer International Publishing.
- Ritter, A., Wright, E., Casey, W., & Mitchell, T. (2015). Weakly supervised extraction of computer security events from twitter. In *Proceedings of the 24th International Conference on World Wide Web, WWW '15*, (pp. 896–905)., Republic and Canton of Geneva, CHE. International World Wide Web Conferences Steering Committee.
- Sayyadiharikandeh, M., Varol, O., Yang, K.-C., Flammini, A., & Menczer, F. (2020). Detection of novel social bots by ensembles of specialized classifiers. *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*.
- Tikhomirov, M., Loukachevitch, N., Sirotina, A., & Dobrov, B. (2020). Using bert and augmentation in named entity recognition for cybersecurity domain. In Métais, E., Meziane, F., Horacek, H., & Cimiano, P. (Eds.), *Natural Language Processing and Information Systems*, (pp. 16–24)., Cham. Springer International Publishing.
- Ural, O. & Acarturk, C. (2021). Automatic detection of cyber security events from turkish twitter stream and newspaper data. (pp. 66–76).
- Wang, D., Al-Rubaie, A., Clarke, S. S., & Davies, J. (2017). Real-time traffic event

- detection from social media. *ACM Trans. Internet Technol.*, 18(1).
- Yagcioglu, S., Seyfioglu, M. S., Citamak, B., Bardak, B., Guldamlasioglu, S., Yuksel, A., & Tatli, E. I. (2019). Detecting cybersecurity events from noisy short text. *CoRR*, *abs/1904.05054*.
- Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R., & Le, Q. V. (2020). Xlnet: Generalized autoregressive pretraining for language understanding.
- Örs, F. K., Aydın, M., Boğatarkan, A., & Levi, A. (2021). Scalable wi-fi intrusion detection for iot systems. In *2021 11th IFIP International Conference on New Technologies, Mobility and Security (NTMS)*, (pp. 1–6).



## APPENDIX A

### Twitter Queries

List of queries that were used to create both SUCyber and SUCyberNER datasets are shared here. They include victim name and time periods for the tweet collection and the table can be used to reproduce the dataset with the tweets that are still available. Also, the datasets that we created are shared. <sup>1</sup>

Twitter Queries	
github ddos lang:en until:2018-03-02 since:2018-03-01	OVH ddos lang:en until:2016-09-23 since:2016-09-22
cloudflare ddos lang:en until:2014-02-11 since:2014-02-10	protonmail ddos lang:en until:2018-06-28 since:2018-06-27
KrebsOnSecurity ddos lang:en until:2016-09-23 since:2016-09-21 Krebs ddos lang:en until:2016-09-22 since:2016-09-21	psn ddos lang:en until:2014-12-26 since:2014-12-24 playstation ddos lang:en until:2014-12-26 since:2014-12-24
bank ddos lang:en until:2012-09-19 since:2012-09-18	xbox ddos lang:en until:2014-12-26 since:2014-12-24
bbc ddos lang:en until:2016-01-01 since:2015-12-31	xbox ddos lang:en until:2014-12-03 since:2014-12-02
czech ddos lang:en until:2017-10-25 since:2017-10-23	spamhaus ddos lang:en until:2013-03-19 since:2013-03-17
reddit ddos lang:en until:2013-04-20 since:2013-04-19	bitfinex ddos lang:en until:2017-06-14 since:2017-06-13
hong kong ddos lang:en until:2014-06-20 since:2014-06-17	bittrex ddos lang:en until:2017-11-25 since:2017-11-24
wikipedia ddos lang:en until:2019-09-07 since:2019-09-06	telegram ddos lang:en until:2019-06-13 since:2019-06-12
pirate bay ddos lang:en until:2012-05-17 since:2012-05-16	nasdaq ddos lang:en until:2012-02-15 since:2012-02-14
ea ddos lang:en until:2020-04-16 since:2020-04-15	electrum ddos lang:en until:2019-04-08 since:2019-04-07
blizzard ddos lang:en until:2020-03-19 since:2020-03-18	league of legends ddos lang:en until:2013-12-31 since:2013-12-30
dreamhost ddos lang:en until:2017-08-25 since:2017-08-24	ea ddos lang:en until:2014-01-04 since:2014-01-03
mastercard ddos lang:en until:2010-12-08 since:2010-12-07	steam ddos lang:en until:2014-01-04 since:2014-01-02
visa ddos lang:en until:2010-12-08 since:2010-12-07	steam ddos lang:en until:2016-12-24 since:2016-12-23
paypal ddos lang:en until:2010-12-07 since:2010-12-06	aws ddos lang:en until:2019-10-24 since:2019-10-23
amazon ddos lang:en until:2010-12-09 since:2010-12-08	twitch ddos lang:en until:2014-08-28 since:2014-08-27
github ddos lang:en until:2015-03-28 since:2015-03-27	heroku ddos lang:en until:2016-10-22 since:2016-10-21
godaddy ddos lang:en until:2012-09-11 since:2012-09-10	pinterest ddos lang:en until:2016-10-22 since:2016-10-21
demonoid ddos lang:en until:2012-07-28 since:2012-07-27	paypal ddos lang:en until:2016-10-22 since:2016-10-21
guardian ddos lang:en until:2016-10-22 since:2016-10-21	spotify ddos lang:en until:2016-10-22 since:2016-10-21
airbnb ddos lang:en until:2016-10-22 since:2016-10-21	tumblr ddos lang:en until:2016-10-22 since:2016-10-21
amazon ddos lang:en until:2016-10-22 since:2016-10-21	reddit ddos lang:en until:2016-10-22 since:2016-10-21
bbc ddos lang:en until:2016-10-22 since:2016-10-21	playstation ddos lang:en until:2016-10-22 since:2016-10-21
cnn ddos lang:en until:2016-10-22 since:2016-10-21	xbox ddos lang:en until:2016-10-22 since:2016-10-21
ea ddos lang:en until:2016-10-22 since:2016-10-21	slack ddos lang:en until:2016-10-22 since:2016-10-21
wow ddos lang:en until:2019-09-08 since:2019-09-07 / world of warcraft ddos lang:en until:2019-09-08 since:2019-09-07	

<sup>1</sup><https://github.com/mustafayd/sutect>

## Details of Classification Models

**CNN:** Number of filters is 256 and kernel size is 3. Pre-trained GloVe word embeddings with size of 50 is used.

**BiLSTM:** The dimension of hidden vectors is 64 and dynamic random word embeddings with size of 200 is used.

**BERT:** Pre-trained BERT model named "bert-base-uncased" is used. The details of the model as follows: 12-layer, 768-hidden, 12-heads and 110M parameters. Learning rate is  $2e-5$  and epsilon is  $1e-08$ . The model was fine-tuned in 4 epochs.

**RoBERTa:** Pre-trained RoBERTa model named "roberta-base" is used. The details of the model used as follows: 12-layer, 768-hidden, 12-heads and 125M parameters. Learning rate is  $2e-5$  and epsilon is  $1e-08$ . The model was fine-tuned in 4 epochs.

**XLNet:** Pre-trained XLNet model named "xlnet-base-cased" is used. The details of the model as follows: 12-layer, 768-hidden, 12-heads and 110M parameters. Learning rate is  $2e-5$  and epsilon is  $1e-08$ . The model was fine-tuned in 4 epochs.

## Details of NER Models

Learning rate is  $5e-5$  and early stopping is used with patience of 2 for all models.

**BERT:** Pre-trained BERT model named "bert-base-cased" is used. The details of the model as follows: 12-layer, 768-hidden, 12-heads and 109M parameters.

**RoBERTa:** Pre-trained RoBERTa model named "roberta-base" is used. The details of the model used as follows: 12-layer, 768-hidden, 12-heads and 125M parameters.

**XLNet:** Pre-trained XLNet model named "xlnet-base-cased" is used. The details of the model as follows: 12-layer, 768-hidden, 12-heads and 110M parameters.