Detecting Cybersecurity Events from Noisy Short Text
Supplementary Notes

Semih Yagcioglu, Mehmet Saygin Seyfioglu[*], Begum Citamak, Batuhan Bardak
Seren Guldamlasioglu, Azmi Yuksel, Emin Islam Tatli
STM A.Ş., Ankara, Turkey
{syagcioglu, msaygin.seyfioglu, begum.citamak, batuhan.bardak,
sguldamlasioglu, azyuksel, emin.tatli}@stm.com.tr

Supplementary Notes

In this supplement, we provide the implementation details that we thought might help to reproduce the results reported in the paper.

What about the model hyperparameters?

In Table 1, we provide the hyperparameters we used to report the results in the paper.

Can we download the data?

Yes. Along with this submission, we provide the whole dataset we collected. Nevertheless, due to the restriction imposed by Twitter, the dataset only contains unique tweet IDs. However, the associated tweets can be easily downloaded with the provided tweet IDs. Dataset is available at http://stmai.github.io/cydec

How to reproduce the results?

Here we describe the key steps to recollect data, retrain model and reproduce results on the test set.

- **Step 1:** As mentioned before, researchers can recollect data through provided tweet IDs.
- **Step 2:** After recollecting data, preprocessing, normalization and tokenization tasks are implemented as detailed in Experiments.
- **Step 3:** In order to learn domain-specific word embeddings on the unlabeled tweet corpus, meta embedding encoders are trained by applying word2vec, GloVe and fastText as discussed in Section 2.
- **Step 4:** Contextual embedding encoder is implemented in order to reveal contextual information as mentioned in Section 2.
- **Step 5:** Network architecture combined by CNNs and RNNs is implemented for detecting cyber security related events as detailed in section 2.

Have you used a simpler model?

We favor simple models over complex ones, but for our task, detecting cyber security related events requires tedious effort as well as domain knowledge. In order to capture this domain knowledge, we designed handcrafted features with domain experts to address some of the challenges of our problem. Nevertheless, we also learn to extract features using deep neural networks.

In the Section 3 of the paper, we also provide ablations where we discuss which part of the proposed method adds how much value to the overall success.

Table 1: Selected Hyperparameters

	Hyperparameter	value
general	vector_size	100
LDA	num_topics	40
	update_every	1
	chunksize	10000
	passes	1
w2v & fastText	window_size	5
	min_count	5
	iter	5
	alpha	0.025
GloVe	window_size	5
	no_components	100
	learning_rate	0.01
	epoch_num	10
Autoencoder	nb_epoch	100
	batch_size	100
	shuffle	True
	validation_split	0.1
CRF	learning_rate	0.01
	l2 regularization	1e-2

[*] Corresponding author.

Table 2: Results for Contextual Feature Combinations

Features	Accuracy
All	0.725
NER & LDA	0.705
LDA & IE	0.69
NER & IE	0.71
IE	0.68
NER	0.64
LDA	0.66

Why did you use all of the contextual features?

At first glance, it might seem that we threw everything that we got to solve the problem. However, we argue that providing contextual features is somewhat yielding a better initialization, thus providing a network to converge better local minima. We also tried out different combinations of contextual features, i.e., LDA, NER, IE by training 2 layered fully connected neural net with them and, although marginally, the combination of all yield the best results, see Table 2. We argue that NER is more biased towards making false positives as it does not consider the word order or semantic meaning and only raises a flag when many relevant terms are apparent. However, results prove that NER's features could be beneficial when used in combination with IE and LDA which indicates that NER is detecting something unique that IE and LDA could not.

How to recollect data?

As our goal is to develop a system to detect cyber security events, thus collecting more data is crucial for our task. Hence, using the seed keywords as described in the paper Section 3, even more data can be collected using the Twitter's streaming API over a desired period.

How about annotations?

We expected annotators to discriminate between a cyber security event and non cyber security event. In that regard, we used a team of 8 annotators, who manually annotated the cyber security related tweets. Each annotator annotated their share of tweets individually, and in sum, the team annotated a total of $2K$ tweets. Following the same procedure, it is possible to annotate more data, which we believe to help achieve even better results.

How is the human evaluation done?

We randomly selected 50 tweets and provided this subset to 8 human subjects for evaluation. Each annotator evaluated the tweets independently for his/her share of 50 tweets. Then, we compared their annotations against ground-truth annotations.

What about hardware details?

All computations are done on a system with the following specifications: NVIDIA Tesla K80 GPU with 24 GB of VRAM, 378 GB of RAM and Intel Xeon E5 2683 processor.

What are the most common words?

Word cloud in Fig.1 represents the most common words inside the dataset without seed keys.

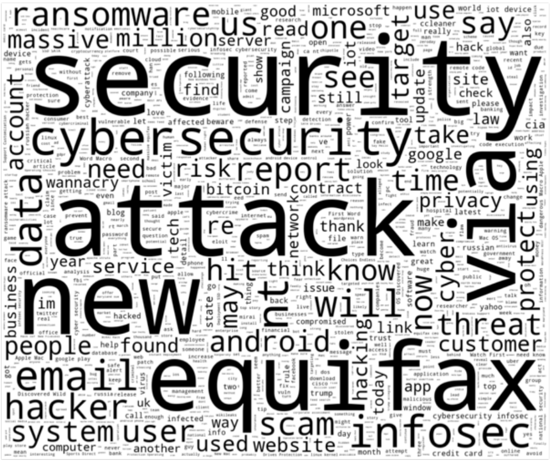

Figure 1: Word Cloud