Automated Extraction of IoT Critical Objects from IoT Storylines, Requirements and User Stories via NLP

SDS2023
IEEE SWISS CONFERENCE ON DATA SCIENCE

uOttawa

Cristovão F. Iglesias Jr
PhD Student at University of Ottawa, Canada
cfrei096@uottawa.ca

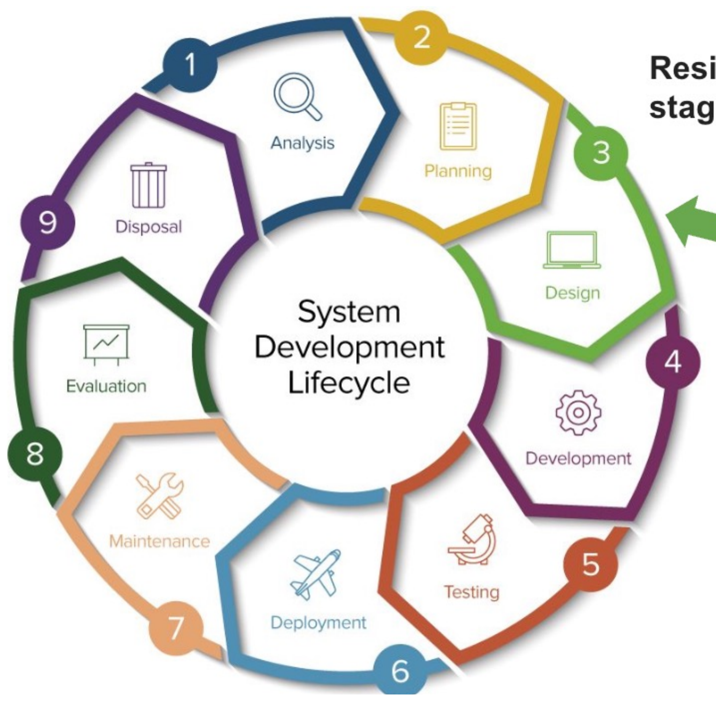
Rongchen Guo
Master Student at University of Ottawa, Canada

Pedro Nucci
Independent Researcher, Poland

Claudio Miceli
Prof. at Federal University of Rio de Janeiro, Brazil

Miodrag Bolic
Prof. at University of Ottawa, Canada

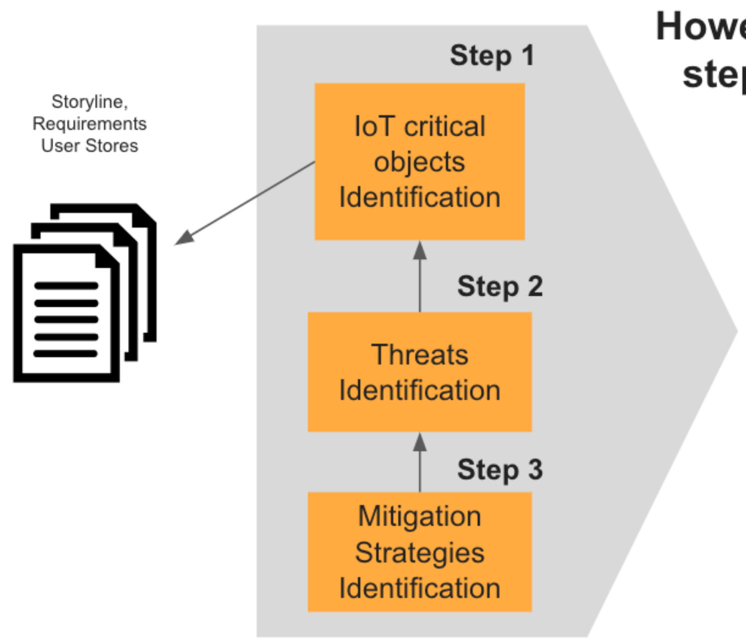
Introduction



Resilience should be addressed in the early stage of IoT application development.

Resilience in design phase allow:

1. Deal with the complexity of the problems,
2. Effective communication,
3. Complete understanding,
4. Reduce costs,
5. Analyze fastly the financial and practical feasibility of the system.



Three steps procedure

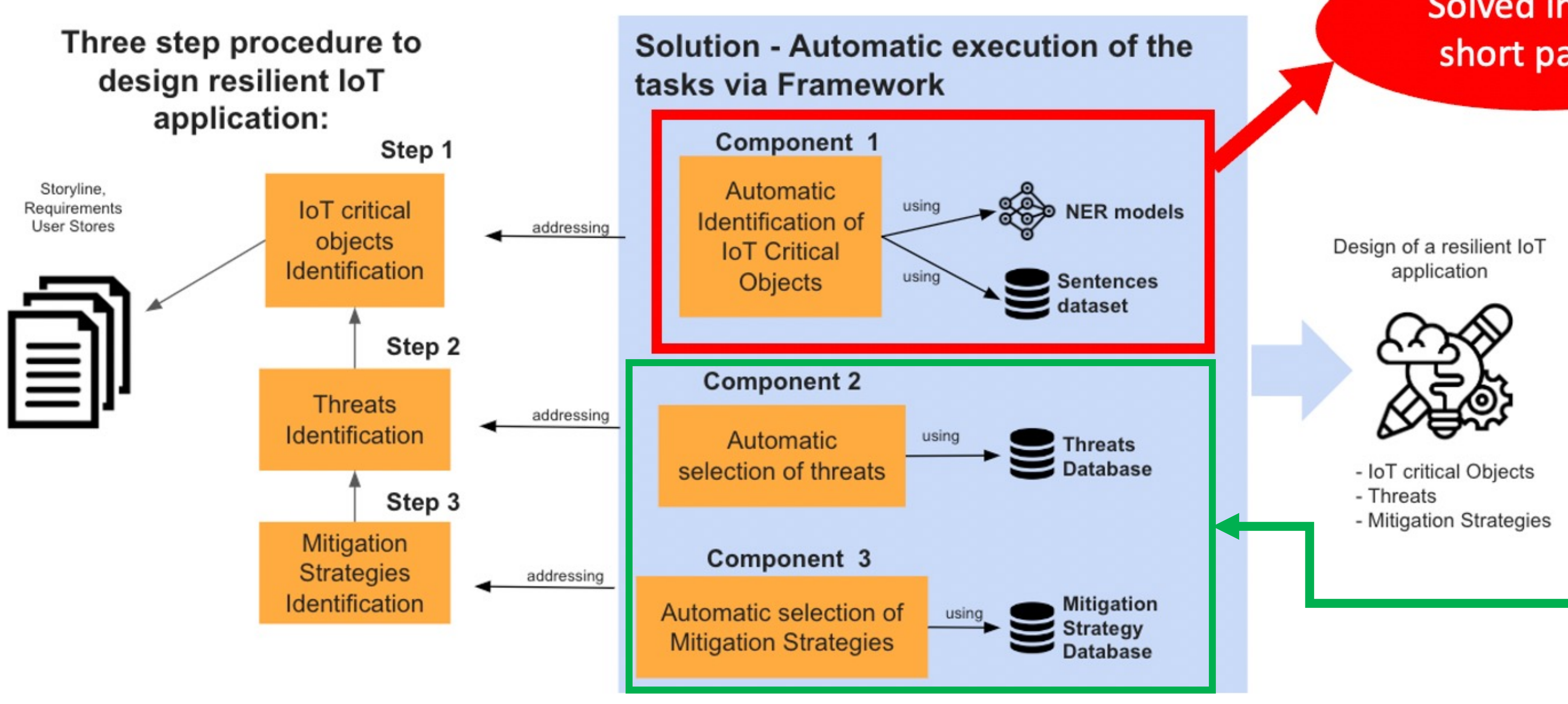
However, the manual execution of three steps procedure to design a resilient IoT application has problems.

Problem

Manual execution of these tasks are:

- **Time-intensive**
- **Prone to error**

Objective



We assessed the usefulness of Named Entity Recognition (NER) models to automatically identify IoT critical objects from documents to make a modelling process faster and less prone to errors.

Technical Approach

Named Entity Recognition (NER) is the process of identifying named entities in text. In our scenario, there are three named entities: Device, Resource, and Service. These IoT critical objects are defined in Table I.

IoT CRITICAL OBJECTS DEFINITION

IoT Critical Object	Definition	Example
Device	Technical physical component (hardware) to monitor or interact with real-world objects.	Mobile Phone, Embedded system, any sensor, actuator, tag or gateway.
Resource	Computational element that gives access to information about, or actuation capabilities on a real-world object.	Device driver, Programming API, Data repositories, data cache on gateway, data on an RFID tag EPCIS repository, ERP database
Service	Software component enabling interaction with resources through a well-defined interface.	Web service or Local service, such as, Alerting, Monitoring, Connecting.

We adopted five deep learning-based architectures for NER in documents.

The five architectures are:

- Pre-trained Spacy [1]
- Pre-trained Bidirectional Encoder Representations from Transformers (BERT) [2]
- Transformers [3]
- Long-Short Term Memory and Conditional Random Fields (LSTM-CRF) [4]
- ELMo (Embedding from Language Models) [5]

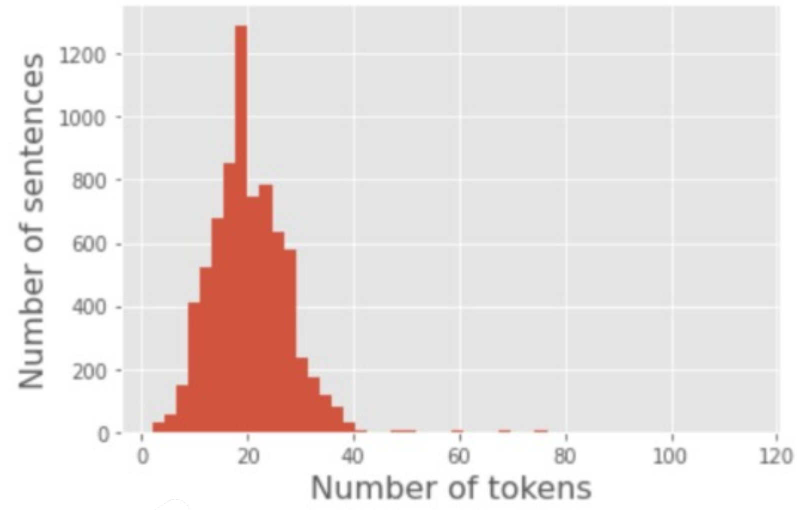
These architectures are the most used for NER applications.

Results

- **Large dataset with 7396 sentences**
- **Three classes of IoT Critical Objects:**
  - **Device, Service and Resource**

DATASET WITH ANNOTATED IOT CRITICAL OBJECTS

		Device	Resource	Service	Total
Dataset Size	Train	1402	2474	931	4807
	Test	755	1332	502	2589
	Total	2157	3806	1433	7396



The biggest sentence has 115 tokens.

NER models and dataset available in: github.com/cristovaoiglesias/iot_critical_obj_extraction_via_nlp

Five NLP architectures:

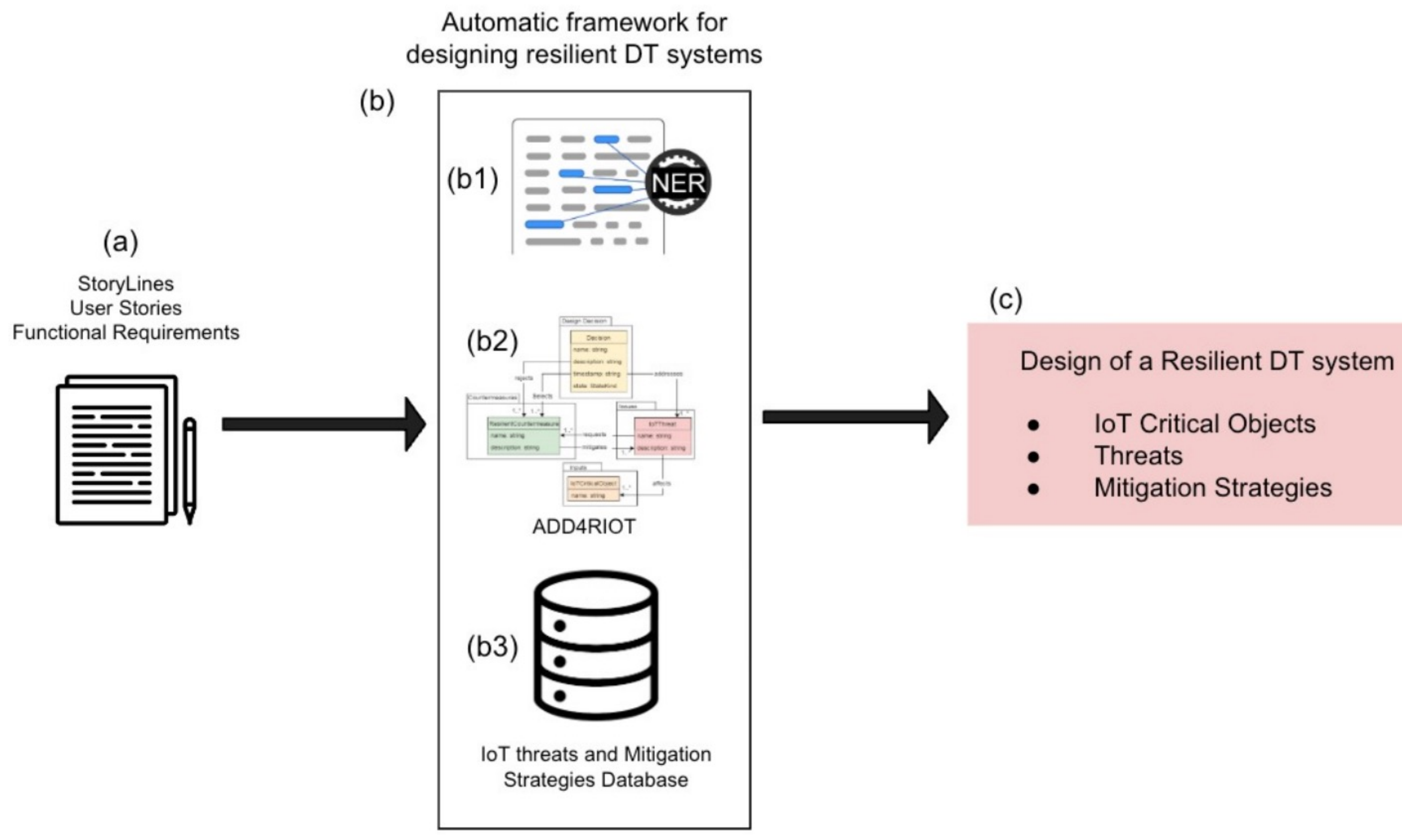
- **Spacy,**
- **BERT,**
- **Transformers,**
- **LSTM-CRF**
- **ELMo**

MODELS PERFORMANCES WITH LARGE DATASET.

Entity	Model	Precision	Recall	F1-score
Device	SPACY	93.9	92.8	93.3
	BERT	93.0	92.0	93.0
	Transformers	92.0	91.0	92.0
	LSTM-CRF	67.0	81.0	74.0
	ELMo	77.0	89.0	83.0
Resource	SPACY	95.8	93.9	94.9
	BERT	90.0	94.0	92.0
	Transformers	96.0	87.0	91.0
	LSTM-CRF	91.0	85.0	88.0
	ELMo	94.0	91.0	92.0
Service	SPACY	87.8	82.5	85.0
	BERT	96.0	88.0	92.0
	Transformers	89.0	67.0	76.0
	LSTM-CRF	88.0	74.0	81.0
	ELMo	95.0	89.0	92.0

Conclusion and Future Works

- A large-sized dataset with annotations regards IoT critical objects
- 5 different NER models to identify IoT critical objects based on concepts of IoT domain model
- The best performance was achieved by fine-tuning the BERT model with the highest F1-score for all entities
- Future works comprise developing a framework to automatically extract IoT critical objects from documents (storyline and requirements) and list all possible IoT threats and resilient countermeasures that can be used in the design of a resilient IoT application.



References

1 - Suppa, Marek, and Ondrej Jariabka. "Benchmarking pre-trained language models for multilingual ner: Traspas at the bsnlp2021 shared task." *Proceedings of the 8th Workshop on Balto-Slavic Natural Language Processing*. 2021.
2 - Devlin, Jacob, et al. "Bert: Pre-training of deep bidirectional transformers for language understanding." *arXiv preprint arXiv:1810.04805* (2018).
3 - Yan, Hang, et al. "TENER: adapting transformer encoder for named entity recognition." *arXiv preprint arXiv:1911.04474* (2019).
4 - Panchendrarajan, Rrubaa, and Aravindh Amaresan. "Bidirectional LSTM-CRF for named entity recognition." *Proceedings of the 32nd Pacific Asia conference on language, information and computation*. 2018.
5 - Ulčar, Matej, and Marko Robnik-Šikonja. "High quality ELMo embeddings for seven less-resourced languages." *arXiv preprint arXiv:1911.10049* (2019).