Security Requirements Classification by Means of Explainable Transformer Models

1st International Workshop on Security and Risk in Identity Management

Luca Petrillo[1,2]
Fabio Martinelli[3]
Antonella Santone [4]
Francesco Mercaldo [4, 2]

1 - IMT School for Advanced Studies Lucca, Lucca, Italy

2 - Institute for Informatics and Telematics of CNR, Pisa, Italy

3 - Institute for High Performance Computing and Networking of CNR, Rende, Italy

4 - University of Molise, Campobasso, Italy

1

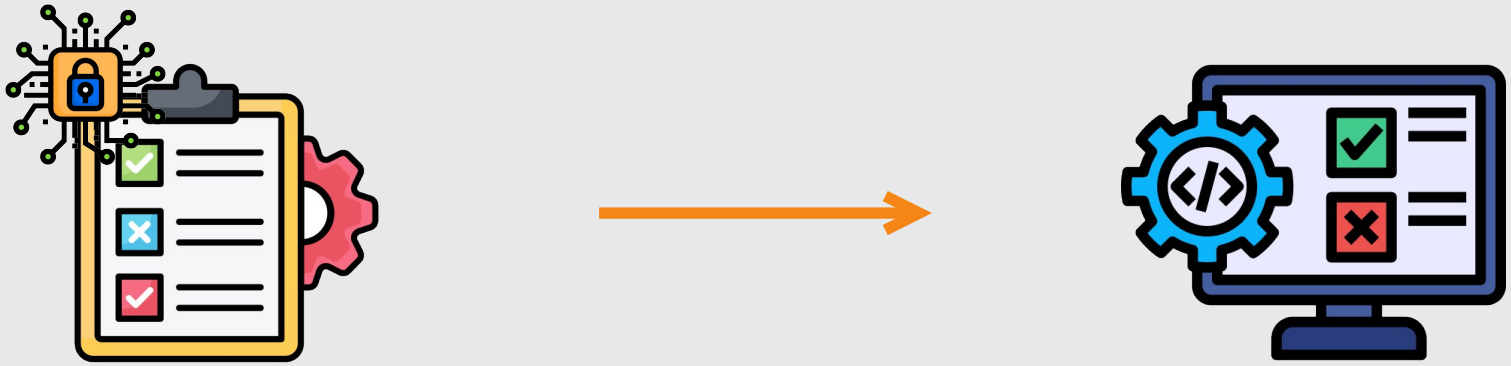
INTRODUCTION

Software requirements for a system are the **description** of what the system **should do**, the **service**(s) that it **provides** and the **constraints** on its operation

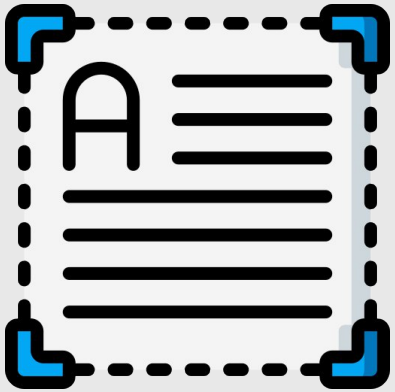
Software requirements can broadly be grouped into two categories: **security** requirements and **non-security** requirements.

Security requirements are **used** to address the measures to **protect** the system from unauthorized access, data breaches, and other security threats

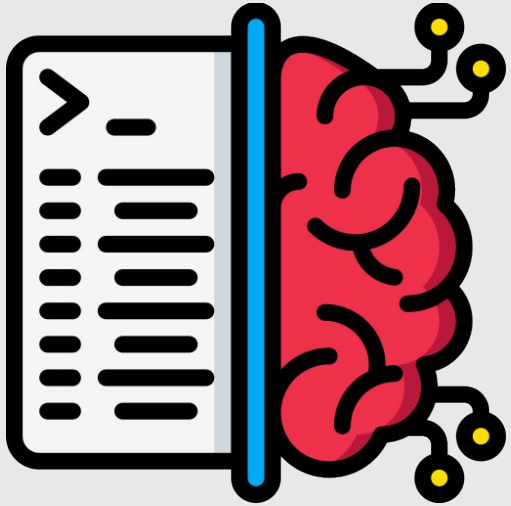
Automating the **identification** process **ensures** that **security** considerations are **integrated** into the software **development** process from the beginning



The problem:



The solution:



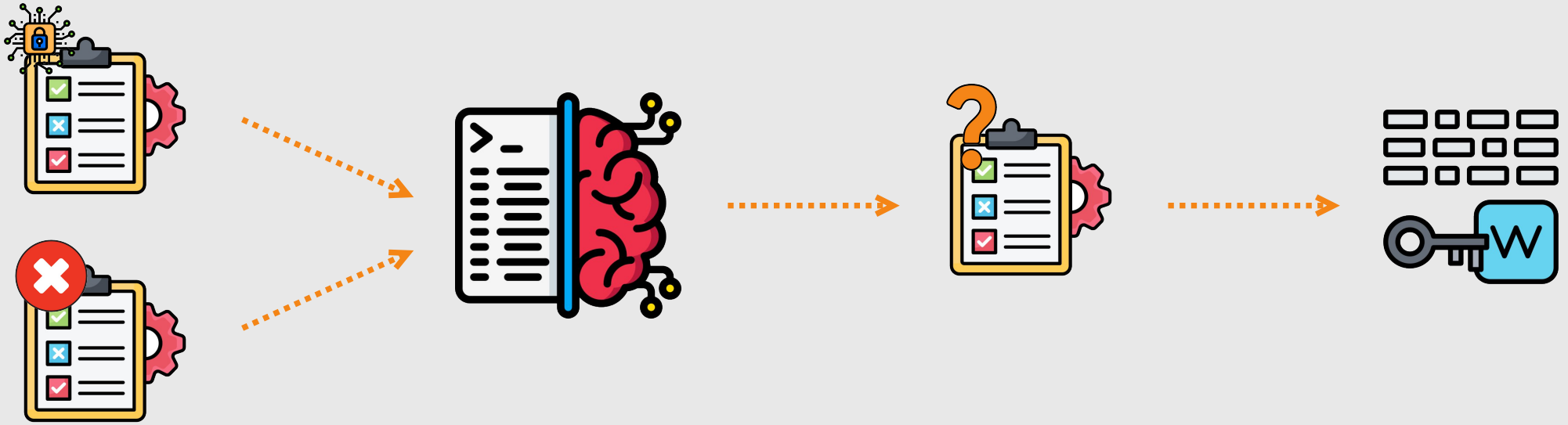
Transformer models

The idea:

2

METHOD

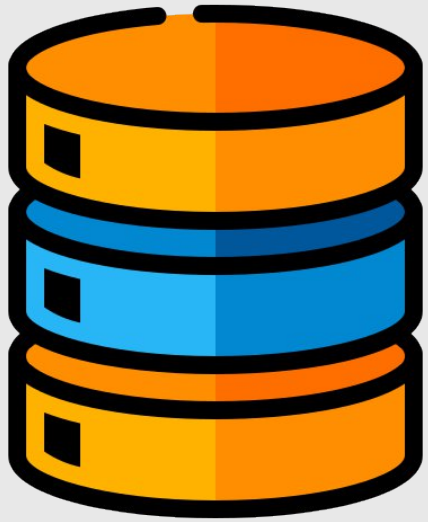
The method:



Software requirements **Transformers fine-tuning** **Evaluation** **Explainability**

2.1 DATASET

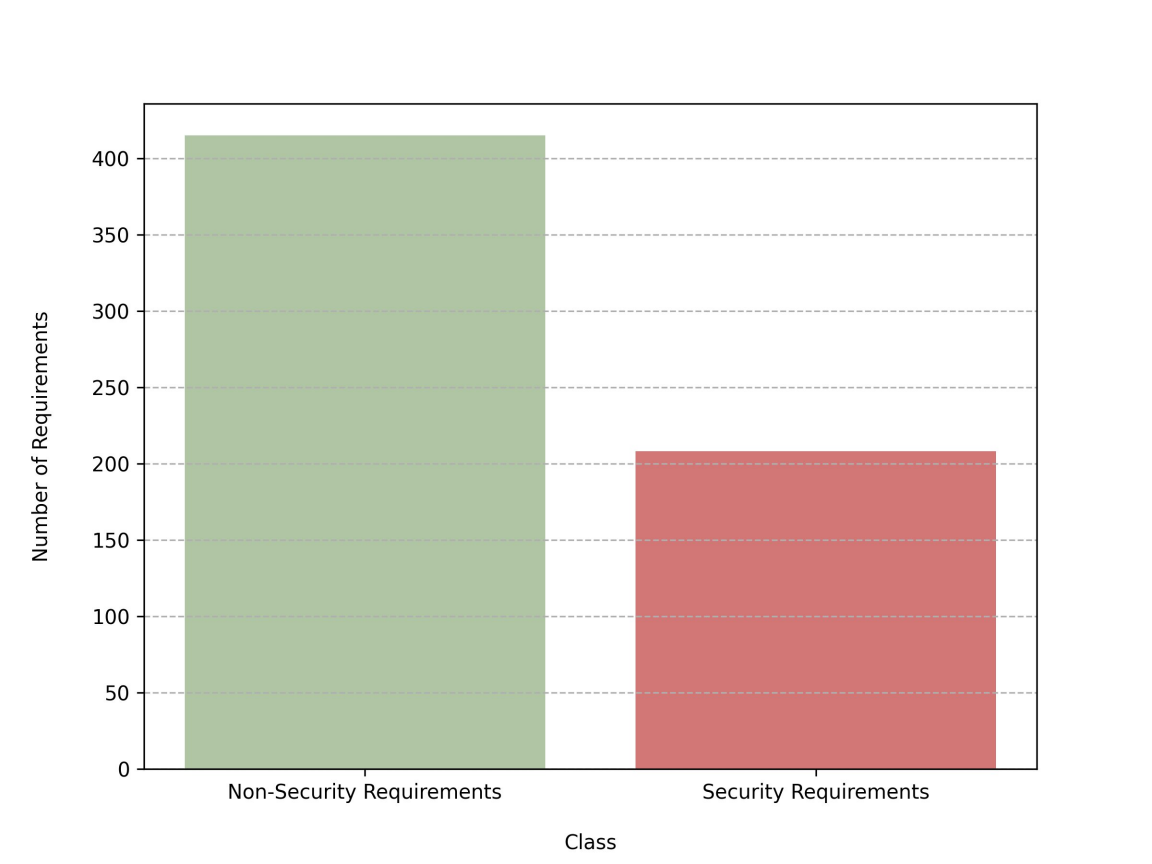
The dataset:



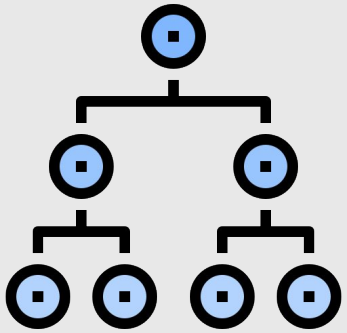
SecReq

E. Knauss, S. Houmb, K. Schneider, S. Islam, and J. Jürjens, “Supporting requirements engineers in recognising security issues,” in Requirements Engineering: Foundation for Software Quality: 17th International Working Conference, REFSQ 2011, Essen, Germany, March 28-30, 2011. Proceedings 17. Springer, 2011, pp. 4–18.

The dataset:

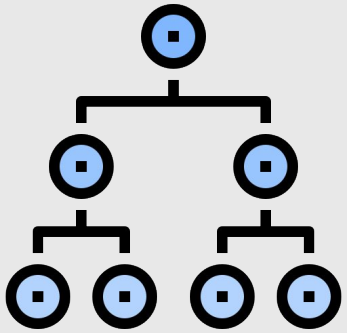

400
350
300
250
200
150
100
50
0
Number of Requirements
Non-Security Requirements
Security Requirements
Class

Pre-processing:



Tokenization

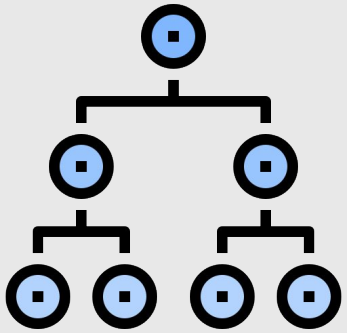
Pre-processing:



Tokenization

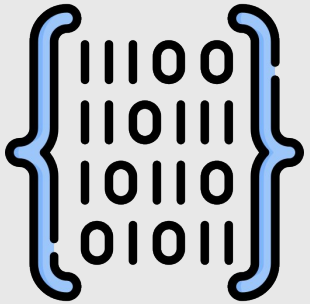
Padding

Pre-processing:



Tokenization

Padding



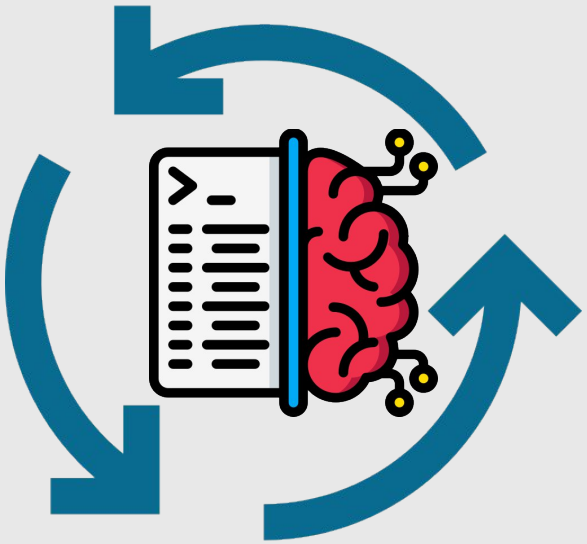
Encoding

Dataset split:

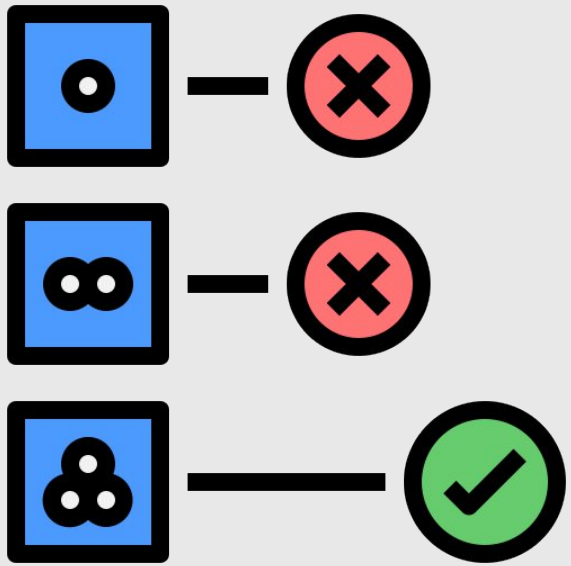
Type	N. of security	N. of non-security	Total
Train	166	331	497
Test	21	41	62
Validation	21	43	64

2.2

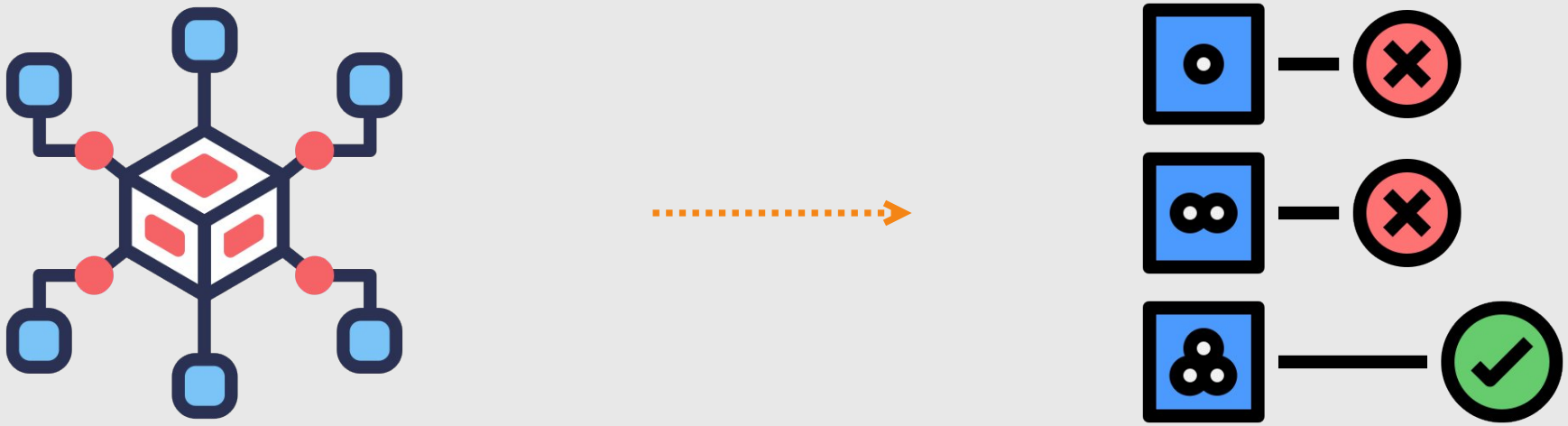
EXPERIMENTS AND DATA ANALYSIS



Hyperparameter tuning:



Hyperparameter tuning:



Hyperparameter tuning results:

Model	Max. F1-Score	Trial
BERT-base-uncased	0.90	36
DistilBERT-base-uncased	0.87	0
DistilRoBERTa-base	0.87	2
RoBERTa-base	0.90	21

Hyperparameter tuning results:

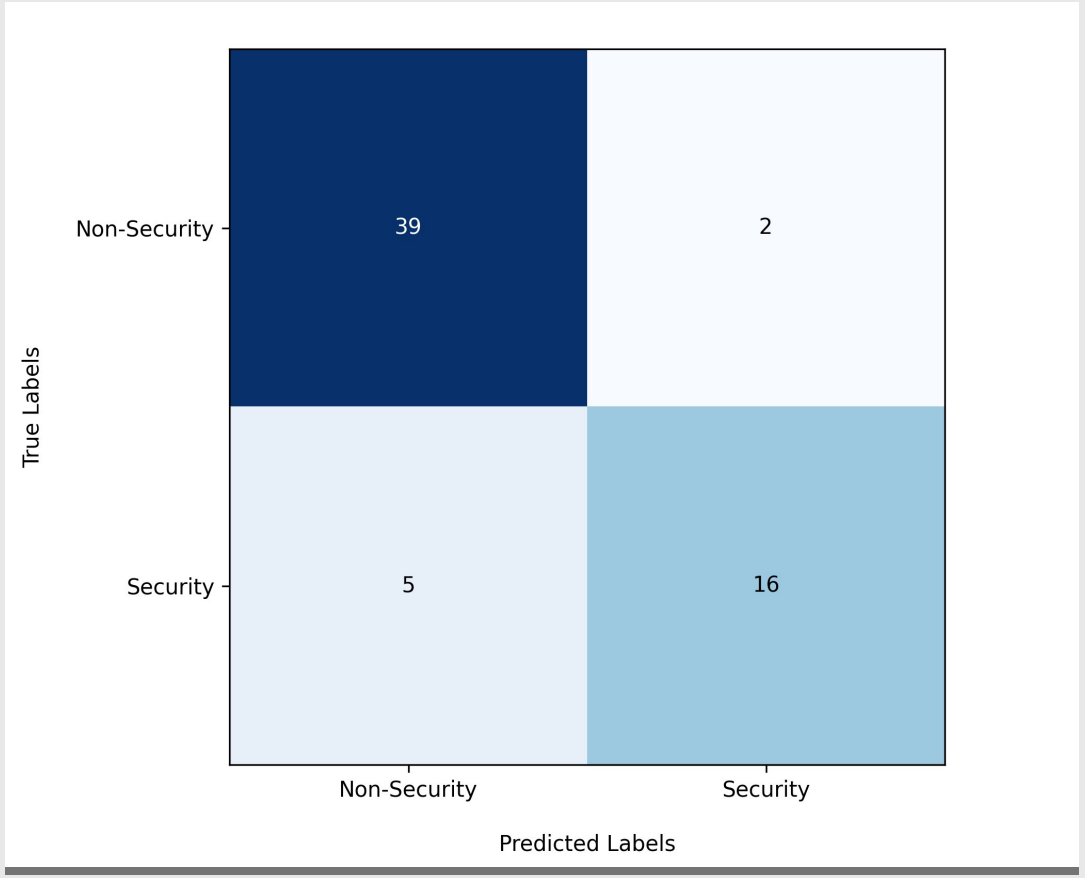
Model	Max. F1-Score	Trial
BERT-base-uncased	0.90	36
DistilBERT-base-uncased	0.87	0
DistilRoBERTa-base	0.87	2
RoBERTa-base	0.90	21

Fine-tuning results:

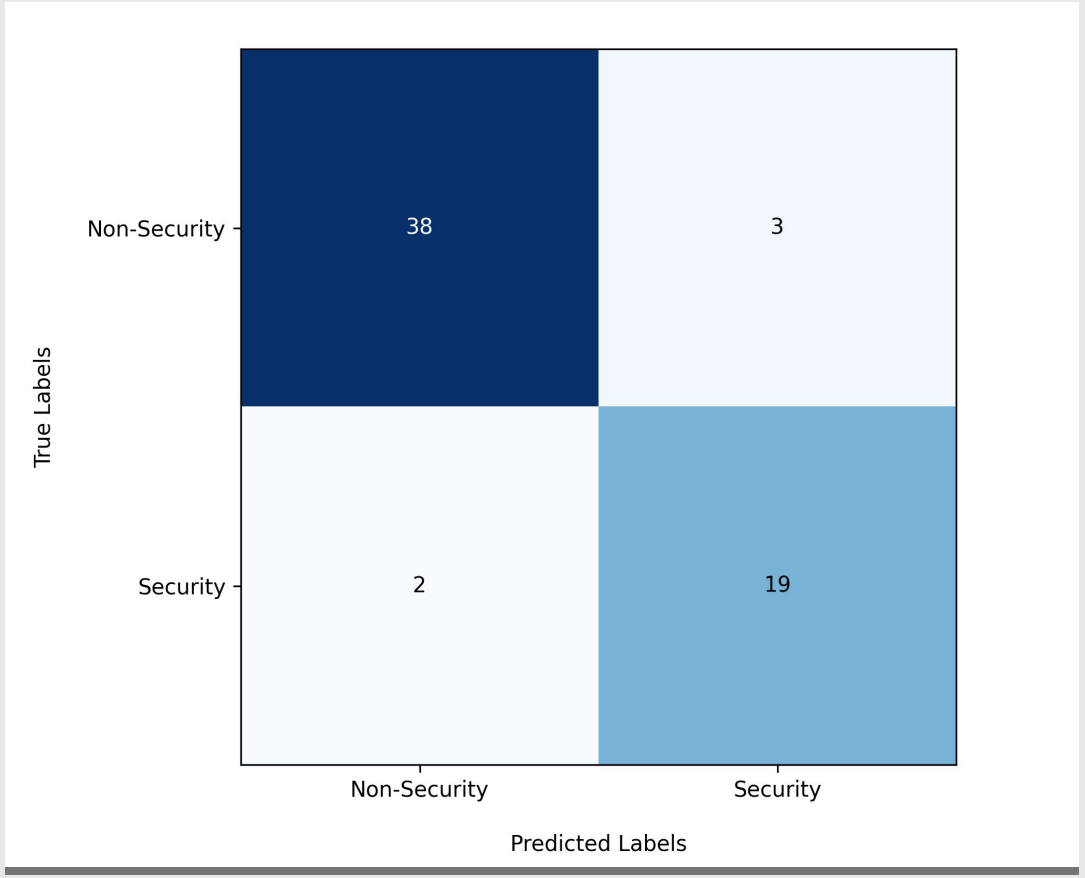
Model	Accuracy	Precision	Recall	F1-Score
BERT-base-uncased	0.89	0.89	0.76	0.82
RoBERTa-base	0.92	0.86	0.90	0.88

Fine-tuning results:

Model	Accuracy	Precision	Recall	F1-Score
BERT-base-uncased	0.89	0.89	0.76	0.82
RoBERTa-base	0.92	0.86	0.90	0.88


Non-Security
Security
True Labels
39
2
5
16
Non-Security
Security
Predicted Labels

BERT

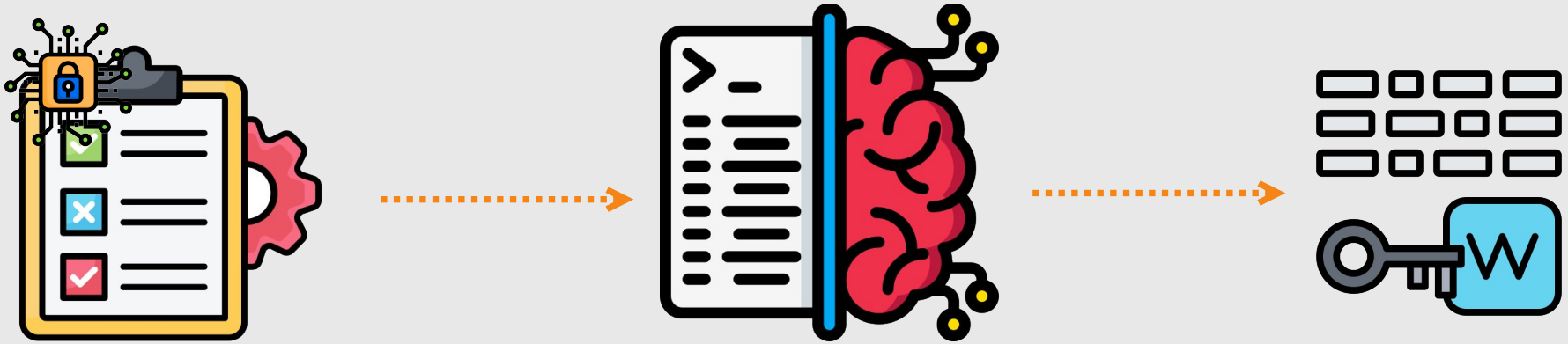

Non-Security
Security
True Labels
38
3
2
19
Non-Security
Security
Predicted Labels

RoBERTa

2.3

EXPLAINABILITY

Explainability:



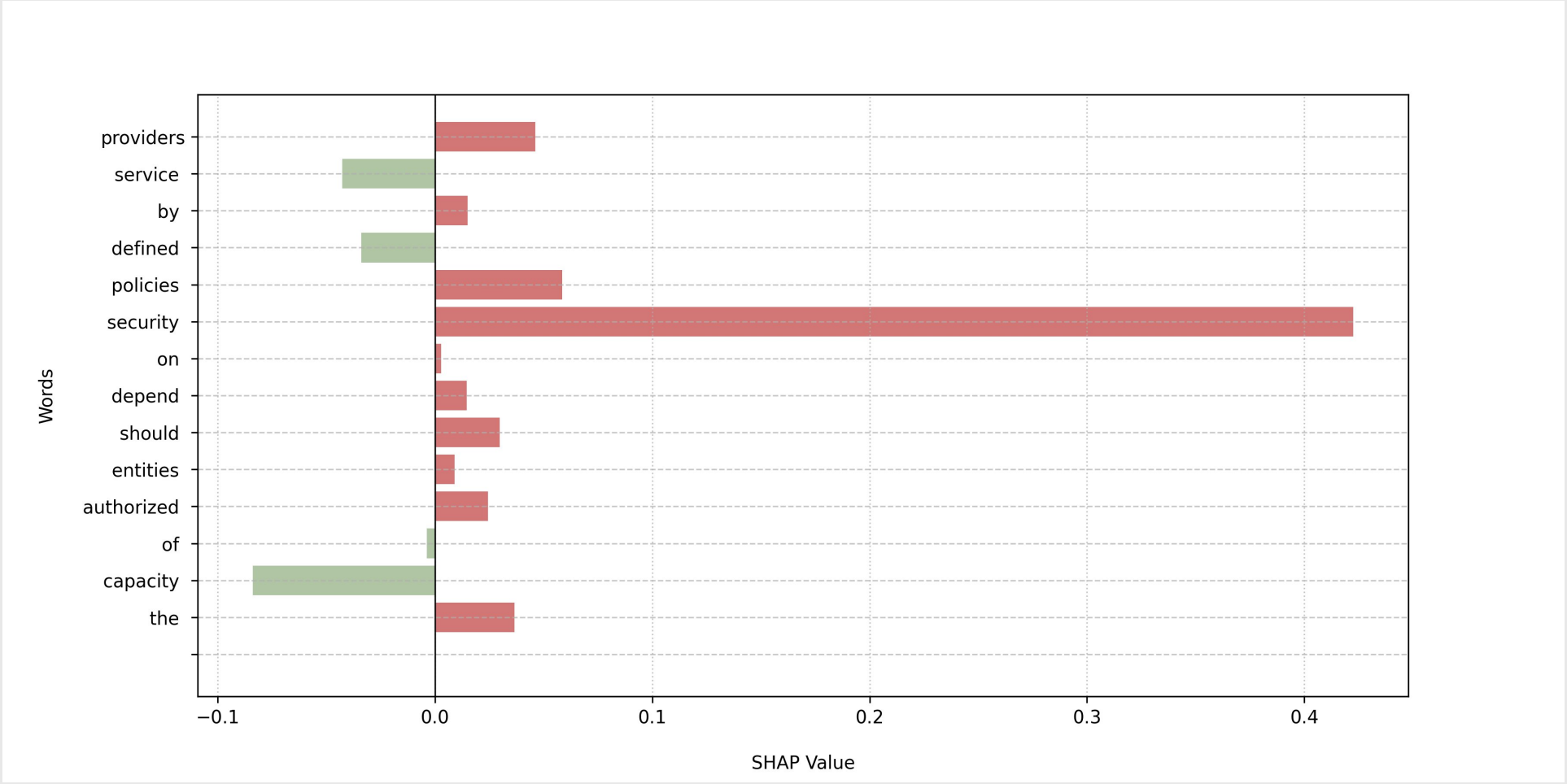
Security requirement

Transformer

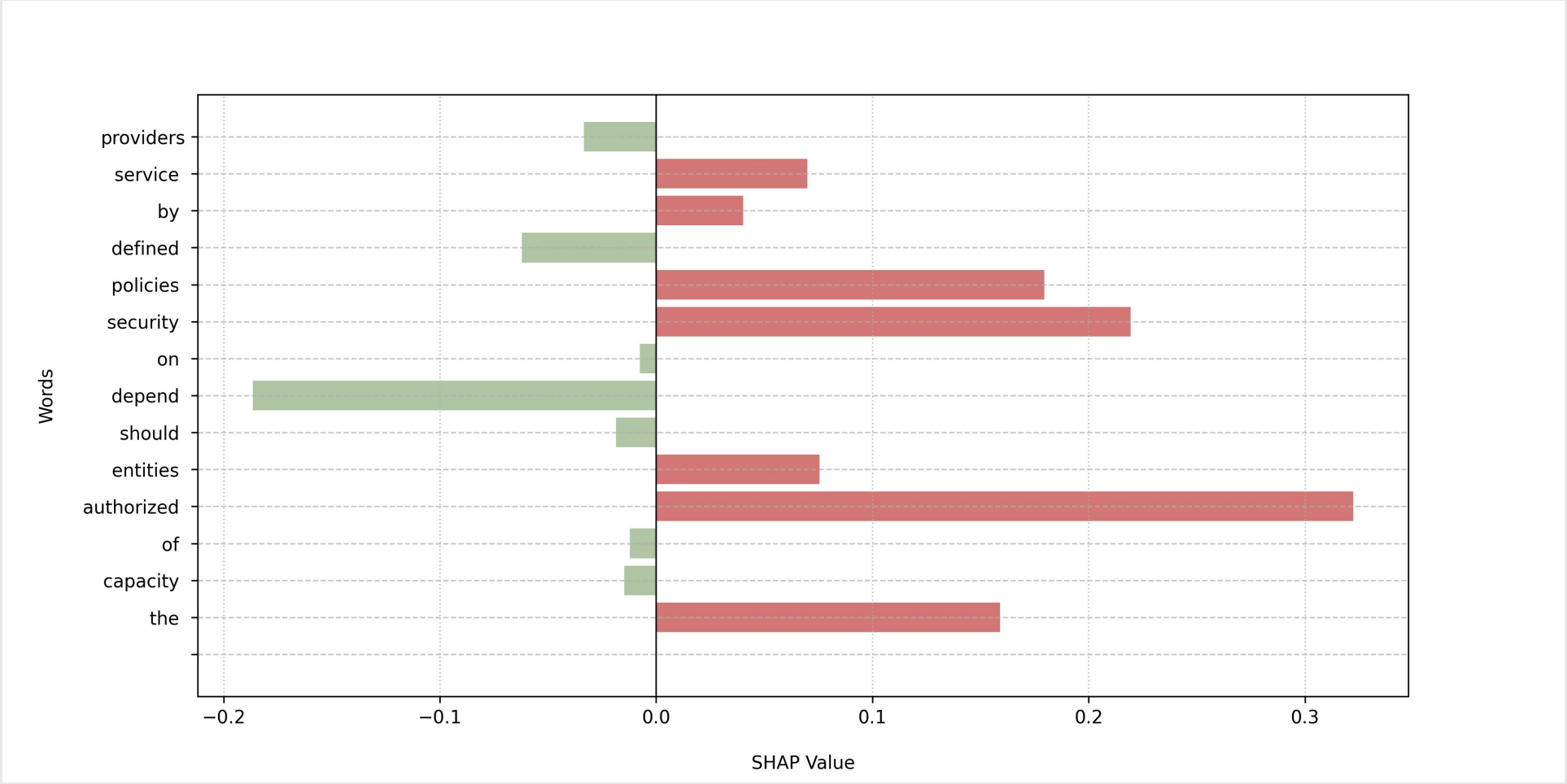
SHAP values

Explainability:

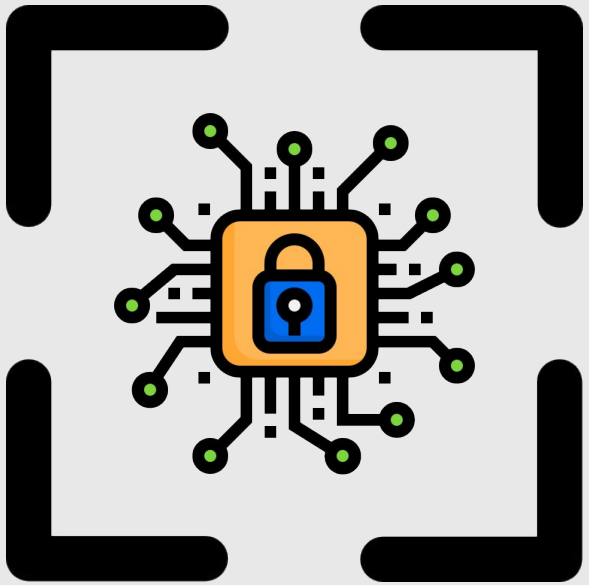
Requirement	True Label	Model	Predicted Label	Score
the capacity of the authorized entities should depend on the security policies defined by the service providers	Security	BERT-base-uncased	Security	0.949
		RoBERTa-base	Security	0.998


providers
service
by
defined
policies
security
on
depend
should
entities
authorized
of
capacity
the
Words
−0.1
0.0
0.1
0.2
0.3
0.4
SHAP Value

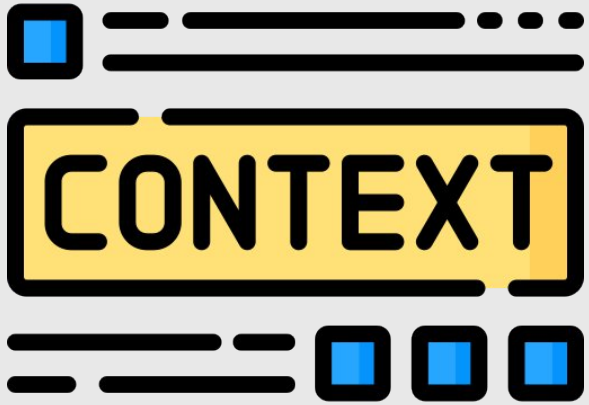
BERT


providers
service
by
defined
policies
security
on
depend
should
entities
authorized
of
capacity
the
Words
−0.2
−0.1
0.0
0.1
0.2
0.3
SHAP Value

RoBERTa



VS



BERT

RoBERTa

4

CONCLUSION AND FUTURE WORK

CONCLUSION - BRIEF RECAP

1. We introduced a method for the automatic identification of security requirements
2. We presented a comparison of various large language models for the classification task
3. We introduced an explainability method to understand the decision-making process

FUTURE WORK

1. Address dataset imbalance by using different techniques
2. Explore other transformer architectures
3. Explore and compare different explainability techniques

THANK YOU!