

# FERMI

## FAKE NEWS RISK MITIGATOR

**Project acronym:** FERMI  
**Project full title:** Fake nEws Risk Mitigator  
**Call identifier:** HORIZON-CL3-2021-FCT-01  
**Start date:** 01/10/2022  
**End date:** 30/09/2025  
**Grant agreement no:** 101073980

## *D3.1 Technology Facilitator Package – 1st Version*

**Work package:** 3

**Version:** 1.0

**Deliverable type:** R - Document, report

**Official submission date:** M16

**Dissemination level:** Public

**Actual submission date:** M22 (revised)



**Leading author(s):**

Surname	First name	Beneficiary
Aziani	Alberto	UCSC - TRANSCRIME
Lo Giudice	Michael Victor	UCSC - TRANSCRIME

**Contributing partner(s):**

Surname	First name	Beneficiary
Dimakopoulos	Nikos	ITML
García Gómez	Joaquín	ATOS
Glöcker	Paul	BIGS
Gousetis	Nikos	INTRA
Papadakis	Thanasis	INTRA
Rieckmann	Johannes	BIGS
Rojas Delgado	Jairo	ATOS
Shadman Yazdi	Ali	UCSC - TRANSCRIME
Shcharbakova	Hanna	UCSC - TRANSCRIME
Stamatis	Giorgos	ITML
Troulitaki	Petrina	ITML
Varela da Costa	Joao	INOV
Vourtzoumis	Michalis	INTRA

**Peer reviewer(s):**

Surname	First name	Beneficiary
Sven-Eric	Fikenscher	BPA
Arttu	Forsell	FMI

**Ethics reviewer:**

Surname	First name	Beneficiary
Giglio	Flavia	KUL

**Security reviewer:**

Surname	First name	Beneficiary
Mattes	Tobias	BPA

## Document Revision History

Version	Date	Modifications Introduced	
		Modification Reason	Modified by
0.05	28.08.2023	Table of contents formulated	UCSC - TRANSCRIME
0.15	06.10.2023	Provision of WP3 partners' first contributions	ATOS, ITML, INTRA, UCSC - TRANSCRIME
0.25	13.10.2023	Revisions to WP3 partners' first contributions	UCSC - TRANSCRIME
0.35	06.11.2023	Second provision of WP3 partner's contributions	ATOS, ITML, INTRA, UCSC - TRANSCRIME
0.45	04.12.2023	Finalized version of WP3 partner's contributions	ATOS, ITML, INTRA, UCSC - TRANSCRIME
0.5	18.12.2023	Contribution by INOV and BIGS added	BIGS, INOV, UCSC - TRANSCRIME
0.6	18.12.2023	Completion of integrated document	UCSC - TRANSCRIME
0.65	26.12.2023	Initial security review	BPA
0.7	07.01.2024	Initial peer review	BPA
0.75	07.01.2024	Initial ethics review	KUL
0.8	18.01.2024	Post-review modifications	UCSC - TRANSCRIME
0.85	18.01.2024	Second security review	BPA
0.9	23.01.2024	Second peer review	BPA, FMI
0.95	29.01.2024	Second ethics review	KUL
1.0	30.01.2024	Final modifications	UCSC - TRANSCRIME
1.1	28.05.2024	Modifications made at PO Request	ATOS, ITML, INTRA, UCSC - TRANSCRIME
1.2	04.07.2024	Final Modifications	UCSC – TRANSCRIME, BPA

## Table of Contents

Executive Summary.....	5
Abbreviations .....	6
1 Introduction .....	7
2 Task 3.1 – The Dynamic Flows Modeler .....	9
2.1 Practical Description .....	9
2.1.1 Estimation and Output .....	9
2.1.2 Understanding the D&FN-Offline Crime Relationship .....	11
2.1.3 Application to Europe .....	11
2.2 Technical Description .....	12
2.2.1 Pre-processing Data .....	12
2.2.2 Machine Learning / AI Architecture .....	17
2.3 Current Advancement and Demo .....	20
2.4 Next Steps .....	20
2.4.1 Left-wing Extremism .....	21
2.4.2 Victim-Author Data and Relationship .....	21
2.4.3 Retraining at Regular Intervals .....	21
3 Task 3.2 – The Spread Analyser .....	22
3.1 Practical Description .....	22
3.1.1 Design .....	22
3.2 Technical Description .....	26
3.2.1 Design .....	26
3.3 Current Advancement and Demo .....	33
3.4 Next Steps .....	34
3.4.1 Functional advances .....	34
3.4.2 Integration of Further Social Media Platforms .....	34
4 Task 3.4 – Swarm Learning for Holistic AI-based Services .....	35
4.1 Practical Description .....	36
4.1.1 The Swarm Learning Framework .....	36
4.1.2 Training ML near to Data Sources .....	39
4.2 Technical Description .....	40
4.2.1 The Fleviden Tool .....	40
4.2.2 A Swarm Learning Solution in Fleviden .....	41
4.2.3 Design Solution’s Main Features .....	44
4.2.4 Preliminary Results .....	45
4.3 Current Advancement and Demo .....	47
4.4 Next Steps .....	48
5 Task 3.6 – The Sentiment Analysis Module .....	49
5.1 Practical Description .....	49
5.2 Technical Description .....	49
5.2.1 The Training Phase .....	49
5.2.2 Inference Phase .....	59
5.2.3 Challenges and Limitations .....	61
5.3 Current Advancement and Demo .....	62
5.4 Next Steps .....	62
6 Integration with Tasks 3.3 and 3.5 .....	64
6.1 Task 3.5 – The Behaviour Profiler and Socioeconomic Analyser .....	64
6.2 Task 3.3 – The Community Resilience Management Module .....	65
7 Conclusion .....	67
References .....	68

## Executive Summary

The D3.1 Technology Facilitators' Package – 1<sup>st</sup> Version aims to provide an in-depth overview of the technologies being developed by the Fake News Risk Mitigator consortium; specifically, said technologies' current states of development at month 12 (the deadline for the project's second milestone, which includes the requirement to develop “[p]reliminary versions of all technologies and modules”) and month 16 (this deliverable's deadline), and how they were achieved; their adherence to commitments made in the Grant Agreement (Work Package 3, Tasks 3.1, 3.2, 3.4, and 3.6), and the *next steps* that will be taken to ensure their timely completion.

At their current state of development, these technologies provide end-users with a wholistic understanding of a given disinformation or fake news event's origin, content, spread, and impact. Through innovative applications of machine learning and artificial intelligence, as well as swarm learning, ensuring data-privacy is protected, end-users will know if disinformation was produced by a human being or a bot, have access to a network recreation of its online-movement, be provided with an analysis of the content's sentiment, and receive a prediction of changes in offline-crime occurrences following its publication.

T3.1, **the Dynamic Flows Modeler**, is an AI-driven crime prediction device which, utilising big-data, natural language processing and machine learning, generates informed estimates for the impact of an online disinformation or fake news event on the number of offline crime occurrences in NUTS2 regions of Europe. Specifically, the device is comprised of several machine learning architectures, varying based on the crime it is evaluating, that are capable of estimating the change in crime levels in a given area, for a selected number of weeks in the future (ideally, between 6 – 12 weeks), once provided a contemporary disinformation or fake news event.

T3.2, **the Spread Analyser**, consists of three main functionalities that capture the spread of a given disinformation or fake news event, on social media, among other accounts, which of these are most influential, and whether said accounts are controlled by humans or operated by bots. The component, starting from the user-provided post, builds a graph depicting the disinformation spread related to the investigated post, tested using X (formerly Twitter). This process maps how the investigated post was propagated amongst other users and showcases the network of disinformation throughout the platform. Furthermore, for each given post in the graph, the component provides complimentary details on said post and the poster, including the poster's public metrics. Subsequently, the application of machine learning models and graph analysis services produce insights regarding the given post's effect on the network and the user's classification as human controlled or bot operated.

T3.4 involved the construction of a **federated learning** paradigm, characterised by implementing decentralised training of machine learning algorithms. Specifically, T3.4 employed a variant of said methodology, swarm learning. **Swarm learning** allows different providers of data to obtain a common model without needing to share private data with each other, maintaining privacy. In the context of FERMI, this involves the pooling of data from several law enforcement agencies without violating data protection concerns.

T3.6, **the Sentiment Analysis module**, emerges as a valuable component within the FERMI project. Designed to analyse the text of social media posts, found within a network spreading disinformation or fake news, and provide end-users with an accurate summary of said texts' sentiments. The module harnesses the power of the cutting-edge BERT language model to understand and analyse text, even when written with a dynamic and endogenous lexicon of social media platforms.

D3.1 also informs on the **integration between the components above and T3.5 and T3.3**, the **Behaviour Profiler & Socioeconomic Analyser** and the **Community Resilience Management Modeler**. T3.5 aims to quantify likelihood and severity of crimes occurring due to disinformation whose combined terms outputs a measurement of risk. The former, T3.3, the Community Resilience Management Modeler, seeks to support law enforcement agencies in their decisions in regards to countering disinformation online and the potential adverse effects it has on crime and society as a whole. It does so by offering countermeasures, specifically with respect to resource allocation. The integration between these technologies and the tasks centric to D3.1 is, specifically, through the Dynamic Flows Modeler, which provides its output to the Behaviour Profiler & Socioeconomic Analyser. The Dynamic Flows Modeler's output is, thus, the input with which T3.3 and T3.5 operate.

## Abbreviations

<b>API:</b>	Application Programming Interface
<b>ARIMA:</b>	Autoregressive Integrated Moving Average
<b>BERT:</b>	Bidirectional Encoder Representations from Transformers
<b>BFP:</b>	Belgian Federated Police
<b>BPA:</b>	Bavarian University of Public Service
<b>CNN:</b>	Convolutional Neural Network
<b>D&amp;FN:</b>	Disinformation and Fake News
<b>FERMI:</b>	Fake News Risk Mitigator
<b>FL:</b>	Federated Learning
<b>FMI:</b>	Finland Ministry of the Interior
<b>GA:</b>	Grant Agreement
<b>GDP:</b>	Gross Domestic Product
<b>GDS:</b>	Graph Data Science
<b>GRU:</b>	Gated Recurrent Unit
<b>LEA:</b>	Law Enforcement Agency / Agencies
<b>LSTM:</b>	Long Short-Term Memory
<b>MAE:</b>	Mean Absolute Error
<b>ML:</b>	Machine Learning
<b>MLP:</b>	Multilayer Perception
<b>NLP:</b>	Natural Language Processing
<b>NUTS:</b>	Nomenclature of Territorial Units for Statistics
<b>PU:</b>	Public
<b>RMSE:</b>	Root Mean Square Error
<b>RNN:</b>	Recurrent Neural Network
<b>SOTA:</b>	State-of-the-Art
<b>SST:</b>	Stanford Sentiment Treebank
<b>TRL:</b>	Technological Readiness Level

### Technologies' Abbreviations:

Task	Grant Agreement Name	Abbreviation in D3.1
T3.1	D&FN-induced and D&FN-enabled offline crimes analysis	Dynamic Flows Modeler
T3.2	Disinformation Sources and Spread Analysis and Impact Assessment	Spread Analyser
T3.4	Swarm learning infrastructure	N/A
T3.6	The sentiment analysis module	Sentiment Analysis module

# 1 Introduction

D3.1, the technology facilitators package, provides an in-depth review of most of the technological components in the Fake News Risk Mitigator (FERMI) platform (those that are covered by Tasks T3.1, T3.2, T3.4 and T3.6, to be exact) and said technologies compliance with the Grant Agreement (GA), wherein commitments were made regarding the technologies' development, function, and performance. In turn, this document will provide an update on the current state of those facets (development, function, and performance), as well as a discussion of the next steps that will be taken to further adhere to the GA and increase the quality of product delivered to end-users. In effect, this document reports on the status of FERMI's technological offering. In full compliance with the GA “[p]reliminary versions of all technologies and modules”<sup>1</sup> described herein were available at the Innovation Flame, in month 12, a crucial project milestone. More specifically, four technological components, independent but well integrated, are featured: (T3.1) D&FN-induced and D&FN-enabled offline crimes analysis, henceforth referred to as the Dynamic Flows Modeler; (T3.2) Disinformation Sources and Spread Analysis and Impact Assessment, henceforth the Spread Analyser; (T3.4) swarm learning, for holistic AI-based services in law enforcement agencies (LEA), and (T3.6) the sentiment analysis module. The Dynamic Flows Modeler committed to “evaluate the degree in which the spread of [disinformation and fake news (D&FN)] online impacts on the occurrence of offline crime,”<sup>2</sup> an analysis which “will evaluate the intensity of the relation between the spread of D&FN and offline crimes, the temporal patterns in the relation, [and] the spatial decay of the relation.”<sup>3</sup> Moreover, the Dynamic Flows Modeler is meant to “produce AI-based [estimates] of the most likely spatiotemporal evolution of D&FN-induced and D&FN enabled offline crimes.”<sup>4</sup> Section 2, which focuses on the Dynamic Flows Modeler, will exhibit the successful adherence, by the Dynamic Flows Modeler, to the above mentioned (and smaller technical) GA commitments.

Section 3 will then provide a thorough overview of the Spread Analyser, T3.2, particularly in its commitment to “create a tool that will take as input news already classified as [D&FN] and will be able to trace and map this news to their main actors/accounts which are responsible for creating and spreading the [D&FN] across the network.”<sup>5</sup> Importantly, the Spread Analyser, in accordance with the GA, can classify if the identified actors/accounts are physical persons or bots and assign an influence index to their role/power over the network.

Subsequently, section 4 covers T3.4, the swarm learning infrastructure designed to “provide a scalable software architecture for training Machine Learning models near to the data sources where they are generated.”<sup>6</sup> In other words, through the development of this swarm learning technology, the FERMI platform, particularly tools such as the Dynamic Flows Modeler, is capable of studying past crime data from multiple, independent LEA partners while not violating their privacy and keeping said data on their servers.

Section 5 discusses the Sentiment Analysis Module, which analyses D&FN, specifically in social media posts, to provide end-users a perception of the emotional tone in said posts' content. The Sentiment Analysis Tool, in accordance with the GA, exploits bidirectional encoder representations from Transformers (BERT) while ensuring the anonymisation of the posts, deletion of links, and replacing of emoji characters with corresponding text/keyword. In doing so, by “the classification [of] results of one specific instance are affected by both past and future instances,”<sup>7</sup> providing end-users a wholistic understanding of the content's sentiment.

An important factor in understanding the aforementioned technologies' function, with the greater FERMI platform, is how they are planned to be integrated with the two other tasks of Work Package 3, T3.3 and T3.5, that is, the Community Resistance Management Module and the Behaviour Profiler/Socioeconomic Analyser. These further technologies are downstream, in terms of the flow of data, from the integrated technologies featured here; therefore, how the various outputs of these technologies are passed to T3.3 and

<sup>1</sup> ‘Grant Agreement: Project 101073980 – FERMI – HORIZON-CL3-2021-FCT-01,’ *European Research Executive Agency*, 2021.

<sup>2</sup> Ibid.

<sup>3</sup> Ibid.

<sup>4</sup> Ibid.

<sup>5</sup> Ibid.

<sup>6</sup> Ibid.

<sup>7</sup> Ibid.

---

T3.5, and how they share the broader infrastructure is addressed. These points, in addition to how the integration complies with the GA is well articulated in the sixth section.

These technological offerings serve LEA needs through providing analytical insights via a linkage with social media platforms, as opposed to merely evaluating a social media post unto itself. The advantage of this linkage is that the FERMI platform can provide a grasp as to the movement of a disinformation post through social media, understanding the influence of the accounts interacting with it. In a post-centric analysis, where no linkage with platforms is established, interactions have a two-dimensional appearance, where every interaction carries the same weight. Evidently, this is not the case, as certain users have far greater pull than others. That being said, access to social media platforms, in terms of establishing a data linkage, comes with its own set of challenges. For the purposes of developing the aforementioned technologies, X was relied on as an effective platform for validating the functionality and effectiveness of the tools, particularly due to the numerous labelled datasets necessary for training AI-based models. That being said, other social media platforms are not precluded from being linked to the platform and its tools. As end-user needs require, the platform could be adjusted to function with a set of other, popular social media platforms. This adaptation of the technologies is, of course, dependent on the availability of suitable data from social media platforms (currently, the FERMI consortium is considering different options, as recommended by the General Project Review Consolidate Report). For the applicable technologies, a subsection is dedicated to illustrate how such end-user driven adjustments would be made.



## 2 Task 3.1 – The Dynamic Flows Modeler

The Dynamic Flows Modeler represents an **advancement in the standard practices of contemporary policing** through a machine learning (ML) crime prevention technology. The Dynamic Flows Modeler can make informed, accurate estimates for the impact of D&FN on levels of crime in European nomenclature of territorial units for statistics (NUTS2) following, and with appreciation for, a given D&FN event. Specifically, its estimates are currently possible with two topics of D&FN: COVID-19 and political extremism. Section 2 will detail the development of this technology, including the data used, its collection, and its cleaning; the ML developed to study past events and make future estimates, and, most importantly, the form and content of the output produced.

Section 2 will be structured as follows, subsection 2.1 will provide a practical description of the Dynamic Flows Modeler (i.e., what can the technology do and, in an overall sense, what does it accomplish); subsection 2.2 then provides a technical description of its production and operation; subsection 2.3 informs as to where the current developed technology is, with respect to where it should be at the end of the FERMI project, and subsection 2.4, subsequently, articulates the *next steps* in advancing the Dynamic Flows Modeler.

Work Package 3, T3.1, covered in this section, featured a commitment to acquire micro-level data for real crime occurrences, offline, and on D&FN, for the purpose of evaluating the relationship between D&FN and offline crime. Moreover, the GA states a necessity for the Dynamic Flows Modeler to make AI-based estimates regarding the spatiotemporal occurrences of D&FN-induced and -enabled offline crime in future periods. How these GA commitments are, or will be, fulfilled by the Dynamic Flows Modeler will be touched on throughout all the proceeding subsections.

### 2.1 Practical Description

The Dynamic Flows Modeler, at its current state of development, fulfils the GA's commitment for a device that can evaluate the relationship between offline crime and the spread of online D&FN, as well as being capable of forecasting future crime occurrences, utilising artificial intelligence and the spatio-temporal evolution of offline crime, given changes in online D&FN.<sup>8</sup> As it stands now, the Dynamic Flows Modeler produces, well, 12 week estimates for D&FN's impact on 11 different types of crime, that were selected in view of data availability and a possible nexus to D&FN, given, preferably, 12 weeks of past crime occurrences and intensity of D&FN. The 11 crimes either imply the immediate use of violence or at least appear to imply a certain proneness to violence, which is in line with the FERMI project's intention to examine the ramifications of D&FN-informed *violent* extremism (see below).

Its accuracy depends on the type of crime selected, reflecting the relationship between some offline crime types and online D&FN, and lack thereof with respect to others. Subsection 2.1 will expand on these two aspects, with 2.1.1 covering the forecasting capacity (and how the output is structured) and 2.1.2 explaining the contributions the Dynamic Flows Modeler makes to understanding the relationship between online D&FN and offline crime. 2.1.3 will then explain the Dynamic Flows Modeler's capacity to be applied to the European context.

#### 2.1.1 Estimation and Output

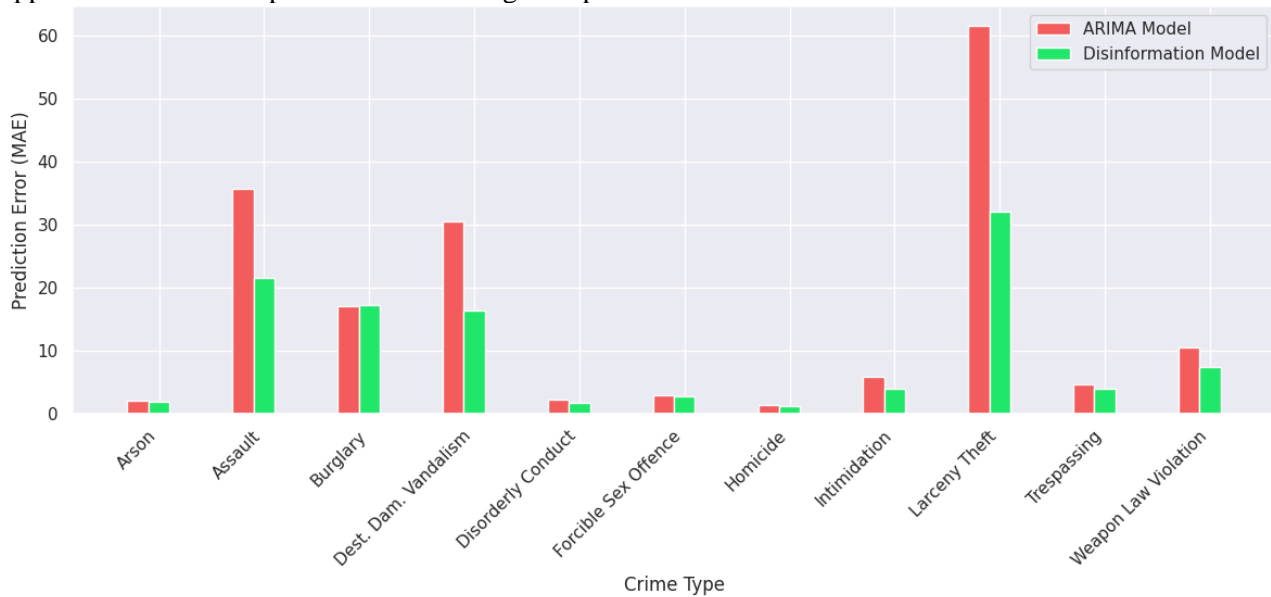
The Dynamic Flows Modeler uses AI-driven ML, particularly deep learning, to study a provided period of time, with its corresponding data, to understand the patterns and evolution of the provided variables, from which it predicts how one of these variables will evolve given the provided evolution of all others. In our case, the Dynamic Flows Modeler is focused on understanding the movement of offline crime (one particular crime at a type) given the socio-economic variables we provided and D&FN's intensity, provided by the platform.

At this moment, the Dynamic Flows Modeler produces estimates for the United States, studying past crime and contextual factors for 31 American places (i.e., 30 municipalities and 1 county) as well as intensity of D&FN spread by and targeted to Americans. All this data sourced between 2018 – 2022, as those were the years in which crime and D&FN data most overlapped, with a sufficient number of observations for the

<sup>8</sup> Ibid.

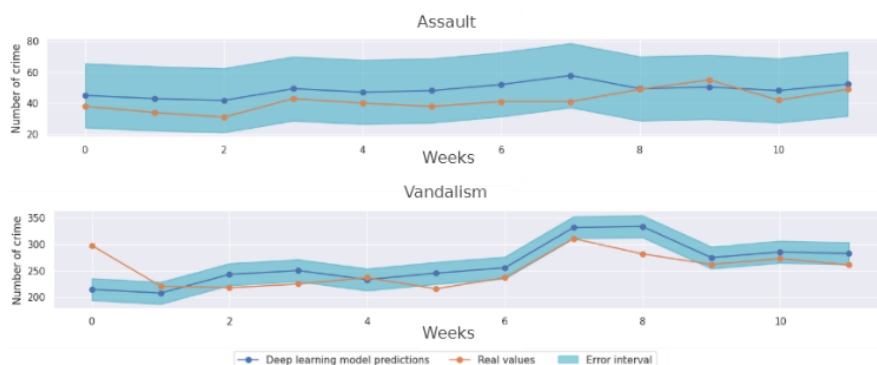
Dynamic Flows Modeler to study. The American context was chosen to overcome a lack of data available for Europe (European data, as pointed out in the General Project Review Consolidated Report, would be preferable,<sup>9</sup> but the use of US data has no major implications for the tool’s functionality and reliability, as clarified below), specifically D&FN data. Datasets of D&FN that met the required nuance (particularly the date of diffusion), quantity, and geo-focus (located in Europe) do not, to the best of our knowledge, exist, let alone are compatible with the FERMI project’s guiding definition of disinformation as 1) factual or misleading nature of the information; 2) intention of the actors to spread such information they know to be false to obtain economic gain or deceive the public; 3) public harm.

That being said, the estimates have proven rather accurate and, as will be explained in 2.1.3, successful application to the European context is being well planned.



**Figure 1: Forecast accuracy by crime type, ARIMA and Dynamic Flows Modeler**

Figure 1 presents the accuracy of the Dynamic Flows Modeler in comparison to an autoregressive integrated moving average model (ARIMA), fed past crime data for the same trial windows. Using the ARIMA as a baseline for state-of-the-art (SOTA) methods available to LEAs, with respect to estimating offline crime, as well as conventional data inclusion to ML estimates of future crime, relying on past crime occurrences, **the Dynamic Flows Modeler represents a significant reduction in mean absolute error (MAE) in all crime types, with rather impeccable accuracy** for certain ones. Figure 1 features the best performing runs of the Dynamic Flows Modeler, in various American places, therefore, it is important to note that for a given European NUTS2 region, for each crime type, forecasts may be of even greater precision<sup>10</sup> or less accurate than presented here.



<sup>9</sup> General Project Review Consolidated Report, p.2.

<sup>10</sup> There is some potential for even greater accuracy once all platform components have been fully integrated and work in tandem, especially as far as the integration of the Dynamic Flows Modeler with the swarm learning framework (see below) and also the spread analyser is concerned.

**Figure 2: Example results for the spatio-temporal evolution of assault and vandalism given D&FN**

### 2.1.2 Understanding the D&FN-Offline Crime Relationship

The relationship between online D&FN and offline crime is, with respect to the academic literature, understood only at a surface level. Scholars have used X's (formerly Twitter) Streaming and Facebook's Graph application program interfaces to investigate if specific instances of online cyberhate would be followed by offline crimes,<sup>11</sup> however, these works consider individual online behaviours, rather than investigating the general spread of D&FN. The Dynamic Flows Modeler, therefore, is rather unique in its commitment to understand the relationship with D&FN spread and offline crime. Its ability to uncover and, in turn, provide insights on the relationship is through its improved capacity in estimating certain crimes' D&FN relationship than others, providing a peek inside the black box.

### 2.1.3 Application to Europe

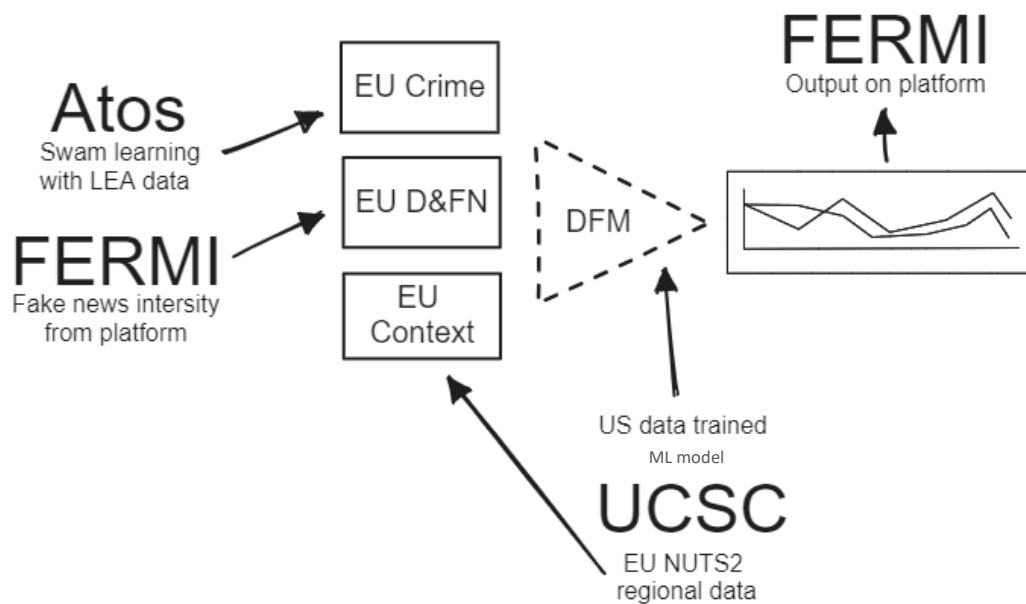
Applying the Dynamic Flows Modeler to European use cases is a matter of providing the technology with the necessary European inputs. To produce a European output, the Dynamic Flows Modeler requires the intensity of D&FN for a window of time prior to  $t_0$ , where  $t_0$  is the time the estimate is being made at. Ideally, this would be a data sequence of 12 weeks, though it does not necessarily need to be. Said D&FN intensity will be provided by the FERMI platform. Moreover, the Dynamic Flows Modeler would require socio-economic data that match the variables it was trained with.

To ensure this is possible, the Dynamic Flows Modeler was trained exclusively with American socio-economic data that had equivalent Eurostat data for the same years, at the level of interest (NUTS2). Lastly, the Dynamic Flows Modeler requires an understanding of the crime occurrences leading into  $t_0$ , which will be provided through FERMI's swarm learning technology, incorporating into the Dynamic Flows Modeler micro-data from LEA while maintaining data privacy meeting the GA commitment to "acquire micro-level data on... actual offline criminal events from participant police authorities."<sup>12</sup> Importantly, while there are some variations between how crimes are defined, between American and European criminal law, the variations are no more substantial than variations between European Union member states and did not represent any significant redefinition.

While there is an evident disconnect between the level of crime in the United States and in Europe, it must be recalled that the Dynamic Flows Modeler studies the relationship between its provided features and offline criminal events, including the criminal events themselves. This means that the Dynamic Flows Modeler takes for granted the existing crime rate of the geo-political space where the training data comes from, and, as previously mentioned, is provided said past crime data for the European countries it is used within. Essentially, the variance between geo-political area of training and use should be mitigated since the Dynamic Flows Modeler understands the movements of the features in conjunction with one another and independently, allowing for a more general use than traditional prediction methods such as seen in econometric analysis.

<sup>11</sup> Burnap, P., et al., 'Hate in the Machine: Anti-Black and Anti-Muslim Social Media Posts as Predictors of Offline Racially and Religiously Aggravated Crime,' *British Journal of Criminology*, 2020; Burnap, P., & Williams, M.L., 'Cyberhate on Social Media in the Aftermath of Woolwich: a Case Study in Computational Criminology and Big Data,' *British Journal of Criminology*, 2016; Gallacher, J.D., et al., 'Online Engagement Between Opposing Political Protest Groups via Social Media is Linked to Physical Violence of Offline Encounters,' *Social Media + Society*, 2021; Muller, K., & Schwarz, C., 'Making America Hate Again,' *SSRN Working Paper*, 2018; Muller, K., & Schwarz, C., 'From Hashtag to Hate Crime: Twitter and Anti-Minority Sentiment,' *SSRN Working Paper*, 2020; Muller, K., & Schwarz, C., 'Fanning the Flames of Hate: Social Media and Hate Crime,' *Journal of European Economic Association*, 2021.

<sup>12</sup> 'Grant Agreement: Project 101073980 – FERMI – HORIZON-CL3-2021-FCT-01,' *European Research Executive Agency*, 2021.



**Figure 3: Process to allow for European application of Dynamic Flows Modeler**

## 2.2 Technical Description

With respect to the methodology used to compose the Dynamic Flows Modeler, two major tasks were undertaken: (1) pre-processing the significantly large collection of data and (2) the development of the ML architecture that produces its forecasts. For the former, natural language processing (NLP), the cleaning of collected crime incidents and of socio-economic controls was necessary for the data to be fitted to the latter, the ML architecture. With the ML, two existing models were reworked for our purposes, convolutional neural network (CNN) and Transformers. Just as well, an ensemble learning method, wherein the two models work together, was built.

### 2.2.1 Pre-processing Data

#### 2.2.1.1 Natural Language Processing

Searching for D&FN to train the Dynamic Flows Modeler was informed by the project’s guiding definition of disinformation: (1) factual or misleading nature of the information; (2) intention of the actors to spread such information they know to be false to obtain economic gain or deceive the public, and (3) public harm. Unfortunately, these definitional building blocks were difficult to find in datasets large enough, in terms of observations, or comprehensive enough, in terms of temporal coverage, to train a ML model. A wide variety of datasets were explored, as reported in table 1, before NELA-GT was decided upon as the best fit.

**Table 1: Considered and excluded D&FN sources for training**

Source Name	Reason for Exclusion
CNN / Daily Mail Complied Dataset	Failed to meet project definition
LOCO	Failed to meet project definition
IRMA	Failed to meet project definition and outside languages of interest
Repository of Fake News	Inaccurate temporal specification
ISOT Fake News	Insufficient observation count
GermanFakeNC	Insufficient observation count
Spanish Fake and Real News	Insufficient observation count
Spanish Fake News Corpus	Insufficient observation count
GRAFN	Insufficient observation count
FakeCovid Fact-Checked News Dataset	Insufficient observation count
LIAR	Insufficient observation count
Kaggle Fake News Dataset	Insufficient observation count
Albanian Fake News Corpus	Insufficient observation count and outside languages of interest
HoaxItaly	Limited temporal range and outside languages of interest
Fakeddit	No geolocation and failed to meet project definition
Fake News Dataset	No temporal specification
WELFake Dataset	No temporal specification
FNC-1	No temporal specification
Snopes Fact-News Data	No temporal specification
FakeNewsNet	Requires extensive X API access
COVID-19 Disinfo Dataset	Requires extensive X API access

Thus NELA-GT was chosen, to serve as the D&FN for training, first put together by researchers from the Technical University of Denmark and Rensselaer Polytechnic Institute, the dataset was updated year after year with the most recent incarnations including contributions by individuals from the University of Tennessee Knoxville. The first edition was published in 2019, provided approximately 800,000 unique articles, the later versions increased to nearly 1.8 million per year. These datasets represent comprehensive coverage of D&FN’s spread in the United States during the past 4 years and are employed throughout SOTA literature on the subject of D&FN. The articles included in NELA-GT were scaled for veracity, allowing for the selection of observations that not only met the first, but also the second and third pillars of FERMI’s disinformation definition.

NLP was used to classify NELA-GT’s articles into FERMI’s three topics of interest: violent extremism rooted in COVID-19 beliefs, violent right- and left-wing extremism.<sup>13</sup> The primary objective of doing so was to then understand the intensity of the spread, for each given topic, through the years 2018 – 2022. Before classification could begin, the articles’ texts were extracted using SQLite Studio client and all texts underwent a standard NLP pre-processing using Python’s natural language toolkit’s library, which involved tokenisation and lemmatisation, as well as the removal of stop words (using the toolkit’s provided stop words) and punctuation. This ensured that the text data was clean and standardised for analysis.

COVID-19 classification was then undertaken by utilising keywords provided in the NELA-GT datasets, specifically, a list of keywords that could be used for extraction.<sup>14</sup> The provided keywords, however, were rather broad, consisting of 241 words. The list was refined to 131 through filtering for relevancy. This refinement was crucial to ensure that only news articles specifically related to COVID-19 were captured, as certain keywords, such as ‘aerosol transmission,’ were used in unrelated contexts. In turn, keyword matching identified articles related to COVID-19 among those labeled by NELA-GT as being from disinformation

<sup>13</sup> Ibid.

<sup>14</sup> Gruppi, M., et al., ‘NELA-GT-2019: a Large Multi-Labelled News Dataset for the Study of Misinformation in News Articles,’ *arXiv preprint*, 2020; Gruppi, M., et al., ‘NELA-GT-2020: a Large Multi-Labelled News Dataset for the Study of Misinformation in News Articles,’ *arXiv preprint*, 2021; Gruppi, M., et al., ‘NELA-GT-2021: a Large Multi-Labelled News Dataset for the Study of Misinformation in News Articles,’ *arXiv preprint*, 2022.

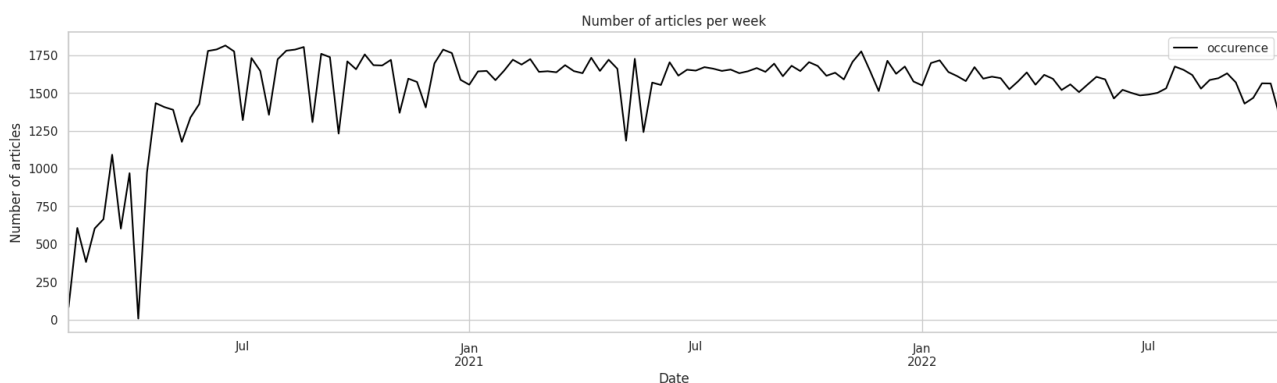
sources. These articles were classified as being COVID-related if they had at least one of the keywords from the refined list.

For right-wing extremism, the methodology differs as there was no NELA-GT provided list of keywords, nor any sufficiently substantial one among the existing literature. We, instead, used manual classification following closely with the Bundesamt für Verfassungsschutz's (Germany's domestic intelligence service's) definition of right-wing extremism, as well as the conceptualisation of the ideology in past academic works.<sup>15</sup> Bundesamt für Verfassungsschutz's defines right-wing extremism as a movement that is extremist and desires a society where there is clear hierarchy, based on the endowed supremacy of certain individuals or the supremacy of a specific sect of society, it should be considered right-wing extremism as the desire for a society where there is clear hierarchy, based on the endowed supremacy of certain individuals or the supremacy of a specific sect of society, achieved through non-democratic means.<sup>16</sup> After several rounds of manual classification, it was concluded that NOQ Report, an American website, represented an abnormally large portion of positives.

Using latent Dirichlet allocation algorithms, keywords were extracted from the universe of NOQ Report articles. Articles were then keyword matched with the NOQ report keyword list, where a threshold of 15 keywords was necessary to be considered as right-wing extremist. Several more rounds of manual classification, still following Bundesamt für Verfassungsschutz (n.d.), Torregrossa (2022), and Botticher (2017), resulted in a list of nine sources being identified as producers of right-wing extremist D&FN.

A similar process was used for left-wing extremism, however, there was no source that published left-wing extremist content at the same rate NOQ Report did for right-wing extremism. The obtained sample size was relatively small and not entirely suitable for comprehensive analysis. Consequently, a decision was made to discontinue further investigation into left-wing extremism at this stage of the project. This allowed for the allocation of resources to other aspects of the classification task and data analysis. Following the Helsinki consortium meeting, in late September 2023, alternative approaches are underway in order to fulfil the GA commitment to study the spatio-temporal relationship between left-wing extremist D&FN online and crime offline.<sup>17</sup>

That being said, intensity was calculated for COVID-19 and right-wing extremist D&FN by considering the number of articles for each respective topic on each day. Intensity was created at a daily level as to allow for varying levels of aggregation depending on the aggregation of crime data. Figures 4 and 5 report their intensities for the years 2020 – 2022.

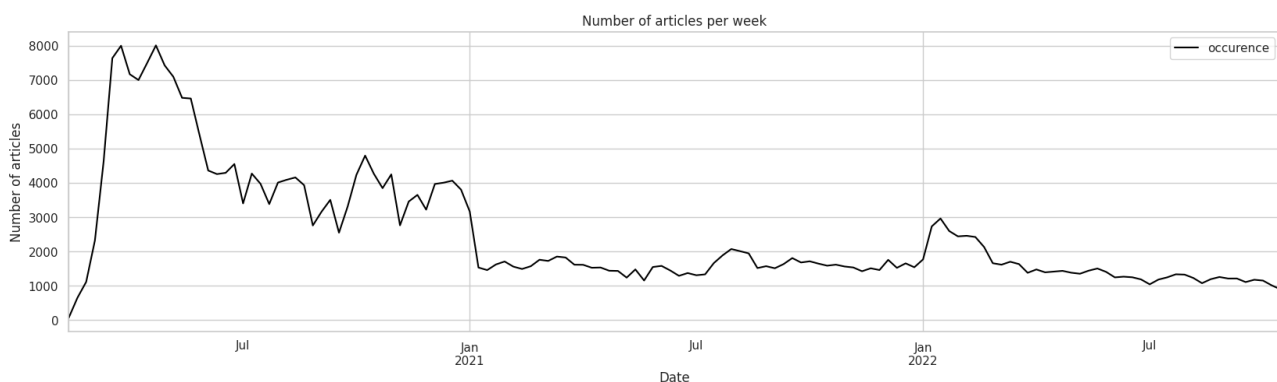


**Figure 4: Right-wing D&FN intensity extracted from NELA-GT (2020 – 2022)**

<sup>15</sup> 'Right-wing Extremism,' *Bundesamt für Verfassungsschutz*, n.d.; Torregrossa, J., et al., 'A Survey on Extremism Analysis using Natural Language Processing: Definitions, Literature Review, Trends and Challenges,' *Journal of Ambient Intelligence and Humanized Computing*, 2022; Botticher, A., 'Towards Academic Consensus Definitions of Radicalism and Extremism' *Perspective Terror*, 2017.

<sup>16</sup> Ibid.

<sup>17</sup> 'Grant Agreement: Project 101073980 – FERMI – HORIZON-CL3-2021-FCT-01,' *European Research Executive Agency*, 2021.



**Figure 5: COVID-19 D&FN intensity extracted from NELA-GT (2020 – 2022)**

There are several potential ethical concerns that could arise from the use of NLP in identifying D&FN. For instance, tokenisation and lammentisation of text content may impact how it is analysed. Moreover, accuracy, with respect to what is D&FN could arise, biasing the analysis. That being said, the NLP undertaken in developing the Dynamic Flows Modeler did not serve to identify articles as being D&FN in any way. Rather, NELA-GT’s academic authors had already classified online sources as being spreaders of D&FN, and, in turn, collected the content they disseminated. NLP was then used by us to classify the D&FN into topics, with which the Dynamic Flows Modeler could study the D&FN-offline crime relationship, as it evolved in the past. This is aligned with overall FERMI platform, which does not identify content as being D&FN, instead, leaving it to the end-user to submit D&FN for analysis.

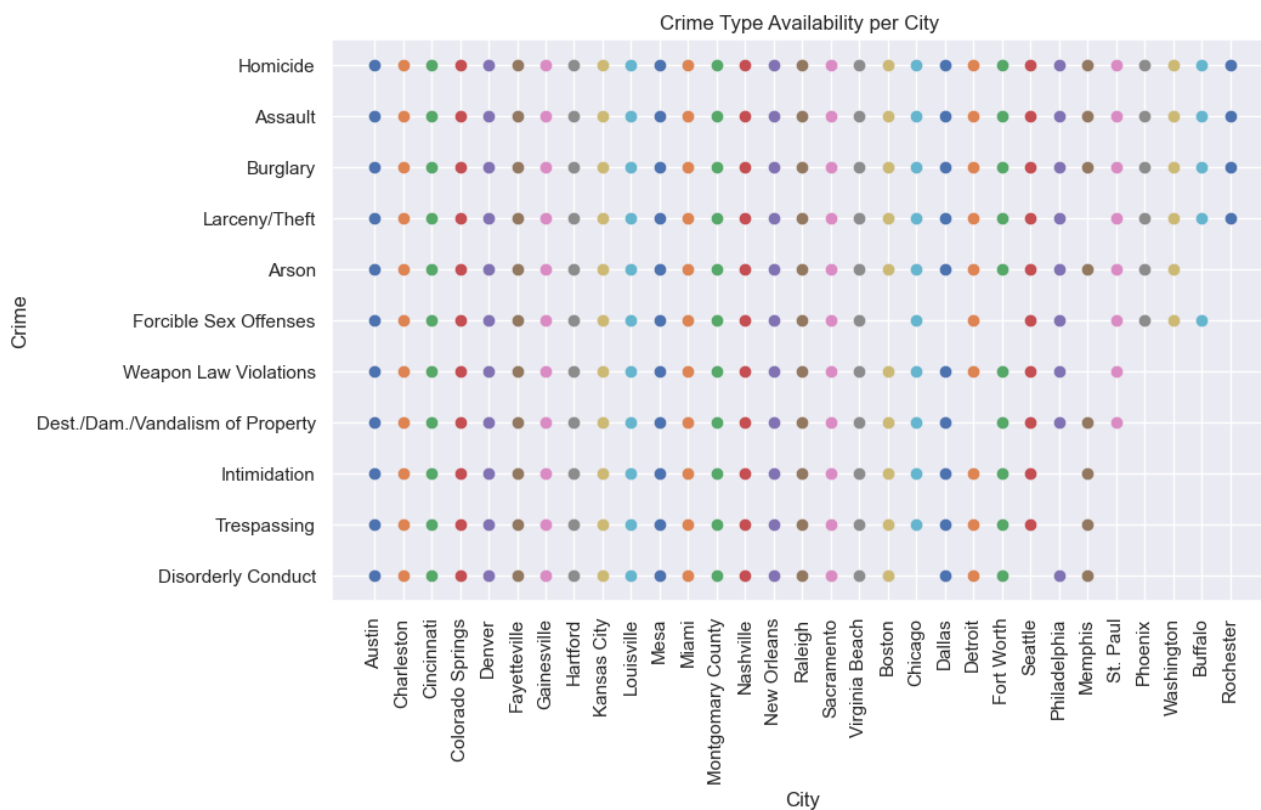
### 2.2.1.2 Collection and Standardisation of Crime Data

As explained above, crime data was collected from 30 American municipalities and 1 county. As further laid out in the preceding remarks, crime data were selected in view of availability and a possible nexus to D&FN. Considering that the FERMI project aspires to examine the ramifications of D&FN-informed violent extremism, it seemed reasonable to pick data on crimes that either imply the immediate use of violence or at least appear to imply a certain proneness to violence. More specifically, the criteria for including a city were based on the format in which they published the data. To be included, the data must have been published at the incident level (i.e., per crime occurrence), with the date of occurrence, and the type of crime that occurred. Just as well, the crime data must have been published publicly (i.e., open source) and for all the years which were covered by NELA-GT (2018 – 2022). Data was also collected for the year 2023, when available, to serve as further unseen observations while we tested the ML models.

The crime types were chosen from the Federal Bureau of Investigation’s universal crime reporting system’s categories. Firstly, financial/white collar crimes (i.e., various forms of fraud, bribery, counterfeiting, embezzlement, and commerce violations) were omitted as they were, theoretically, too far removed from political extremism or COVID-19 related D&FN. Secondly, several ‘victimless’ crimes were excluded, as they did not fit the objective of FERMI (to study the ramifications of D&FN in the field of violent extremism) nor was there academic literature to support a relationship between said crimes and D&FN; specifically, these were gambling, fugitive, immigration, pornography, treason, loitering, drunkenness, perjury, and non-violent family offenses. Non-forcible sex offenses (incest, statutory rape, and failure to register as a sex offender), were likewise omitted due to lack of supporting evidence for a D&FN connection, as well as a general lack of data availability. Motor vehicle theft, drug offenses, prostitution, criminal road violations, human trafficking, kidnapping, extortion, and animal cruelty were also left out given similar concerns to the abovementioned.

The data was standardised in terms of crime type and date format, ensuring all crime types matched the categories of the universal crime reporting system and the standard American date format of month, day, year as it was more common through the datasets. Due to some municipalities data privacy policy preventing the release of certain crime types (e.g., victim privacy with respect to sexual assault) several datasets were produced, for each crime type, including the instances only from cities which published on said crime type. That is to say, the Dynamic Flows Modeler did not study crime instances from 31 American places for each type of crime, but rather, a varying number; see Figure 6 for the intersection of cities and crime types. Crime

occurrences, within these datasets, were transformed from one crime per row to panel data, with one row being one day and the number of crimes, for each type of crime, as the column.



**Figure 6: Crime types available per American place**

For application to the European context, and in accordance with the GA requirement to “acquire micro-level data on... actual offline criminal events from participant police authorities,”<sup>18</sup> partner LEAs, in the FERMI consortium, provided data on crimes instances within their respective territories. Further, through T3.4 (which will be discussed at greater depth later), data will be collected from LEAs via swarm learning, maintaining the privacy of data while providing past crime occurrences to the Dynamic Flows Modeler.

### 2.2.1.3 Socio-Economic Control Variables

In line with the GA desire for “the understanding of the cultural and societal aspects of D&FN”, the Dynamic Flows Modeler includes the socio-economic context in which collected crime incidents and the D&FN is occurring.<sup>19</sup> For the training of the Dynamic Flows Modeler, this data was collected from American sources, specifically the United States Department of Labor and Census Bureau. For application, European micro-level data was collected from Eurostat or directly from LEA partners. In both contexts, mobility data was sourced from Google’s report on community mobility. All of the processed data was anonymous. Against this backdrop, informed consent or other proceedings under the General Data Protection Regulation to allow the processing of personal data (including personal identifiers) was not required. The variables currently appreciated by the Dynamic Flows Modeler (for both training and application) are as follows: (1) population, (2) Gross Domestic Product (GDP) per capita, (3) gender demographics, (4) age demographics, (5) unemployment rate, (6) educational attainment, (7) law enforcement presence, and (8) spatial mobility.

In the American data context, these variables were collected at the municipal level, which could potentially lead to a bias in the results against certain communities linked to particular socio-economic realities. Due, by-and-large to the lack of distance between the municipal and individual level of analysis. However,

<sup>18</sup> Ibid.

<sup>19</sup> Ibid.



when the Dynamic Flows Modeler is applied to Europe, these statistics are matched with NUTS2 regional values and European past crime data is used, helping to eliminate potential bias against certain communities.

## 2.2.2 Machine Learning / AI Architecture

The Dynamic Flows Modeler does not rely on a single architecture, or single ensemble learning of varying independent architectures. Rather, in the process of creating the Dynamic Flows Modeler, multiple ML/deep learning architectures (and ensembles of them) were employed to identify which worked best for each crime type. The best performing architecture for each type was then included in the final composition of the Dynamic Flows Modeler. In other words, depending on the crime type the estimate is requested for, the Dynamic Flows Modeler will utilise the architecture (or ensemble of architectures) identified as being most accurate during training. The proceeding subsections will outline how the inputted data sequence was modified for use within ML and the two architectures, as well as their ensemble, that comprise the Dynamic Flows Modeler.

### 2.2.2.1 Input

Further modifications were made before inputting the data into the models. To gain deeper insights into long-term trends and mitigate the impact of daily fluctuations, there was an aggregation of the collected, daily data into weekly observations. Moreover, all continuous variables underwent a logarithmic transformation  $\log(x + 1)$  and all ratio variables standardised into a range between 0 and 1.

Multiple distinct models were developed, each tailored to a specific type of crime, resulting in a dedicated model for each crime type. To prepare the data for input into these models, a windowing methodology was employed. This involved segmenting the data into 12-week intervals, with a stride of 12 weeks, ensuring that there was no overlap between consecutive windows. As a result, the input matrix for each model comprises a 12-week data sequence, encompassing details regarding the specific crime category under examination, along with data on D&FN intensity, mobility, and macroeconomic controls corresponding to the respective place for each of those weeks. To enhance the model's understanding of seasonality within each input window, additional features, such as season- and month-based dummy variables were introduced. For the model to capture diverse crime data scales, all windows were scaled to a consistent range. The windows were split into training and testing sets, with an 80% allocation for training data and a 20% allocation for testing data.

Thus, the output of each model consists of up to 12 integer values, each representing the estimated incidences of the specific crime type for a maximum of 12 future weeks. This comprehensive approach ensures a more robust analysis of crime trends, while accounting for seasonal variations and monthly influences, and is in line with the GA, as AI models have been adopted and, as will be explained shortly, have maximised the capabilities of the Dynamic Flows Modeler.<sup>20</sup>

### 2.2.2.2 ARIMA

**As a baseline to judge the Dynamic Flows Modeler's accuracy, an ARIMA model was used.** As a stalwart model for making time-series backed forecasts, ARIMAs study the past assuming the future will resemble it, and, thus, provide an estimate that struggles to understand 'interventions'<sup>21</sup> which we posit come in the form of D&FN. The choice to use it as a baseline is, therefore, sound, as the difference in its accuracy, compared to the Dynamic Flows Modeler's can be interpreted as both being due to the improved methodology in the D&FN's architecture and the inclusion of D&FN, a novel factor in estimating crime occurrences.

<sup>20</sup> Ibid.

<sup>21</sup> Hayes, A., 'Autoregressive Integrated Moving Average (ARIMA) Prediction Model,' *Investopedia*, 2023.

### 2.2.2.3 1-Dimensional Convolutional Neural Networks

CNN is a model proven to be effective in studying time-series data.<sup>22</sup> It typically consists of two fundamental components: the CNN itself, which extracts and filters the relevant features and the fully connected layer, which produces the estimates using said features and their relevance. In effect, the CNN, by studying the past, provides weights to the variables it is provided. To introduce non-linearity into the network, rectified linear unit activation functions are employed in each convolutional layer.<sup>23</sup> The convolutional operation in the  $i - th$  layer of the  $j - th$  set can be represented as follows:

$$Z[i, j] = W[i, j] * X + b[i, j]$$

**Equation 2: Convolutional operation of CNN<sup>24</sup>**

$$A[i, j] = ReLU(Z[i, j])$$

**Equation 3: Rectified linear unit activation of CNN convolutional operation output<sup>25</sup>**

Where  $Z[i, j]$  is the convolutional output, a number expressing the role of a feature in the forecast,  $W[i, j]$  is the weight as a matrix, for the  $i - th$  layer of the  $j - th$  set,  $X$  is the provided input, and  $b[i, j]$  is the bias term meant to offset the activation function.<sup>26</sup>  $A[i, j]$  is then the output, after applying rectified linear unit activation.

The fully connected layer consists of four dense (fully connected) layers. These layers are responsible for further feature refinement and dimensionality reduction. The fully connected part of the network leverages the learned features from the convolutional layers to produce the final output, making it a critical component of the entire architecture for tasks such as regression.

$$Z[k] = W[k] \cdot A[k - 1] + b[k]$$

**Equation 4: Fully connected layer of CNN<sup>27</sup>**

$$A[k] = ReLU(Z[k])$$

**Equation 5: Rectified linear unit activation of CNN fully connected layer output<sup>28</sup>**

Where  $Z[k]$  is the output of the  $k - th$  fully connected layer,  $W[k]$  is the weight as a matrix, for the  $k - th$  fully connected layer,  $A[k - 1]$  is the rectified linear unit activation of the previous fully connected layer,  $b[k]$  is the bias for the  $k - th$  fully connected layer, and  $A[k]$  is the output after applying rectified linear unit activation to the output of Equation 4.

Our CNN design includes 3 convolutional blocks, each consisting filters. The initial set employs 500 filters, followed by 250 filters in the second set, and 128 filters in the third set. These filters apply convolutional operations, enhancing the network's capacity to recognise significant patterns in the data. Rectified linear unit activation was employed in each convolutional layer and the model was trained to minimise the mean squared error loss. While training, the model aims to minimise the mean squared error by continually adjusting the parameters to improve accuracy and lower the discrepancies between estimates and target value. Equation 6 presents how mean square error was calculated, with  $n$  being the total number of data points,  $y_i$  the target value

<sup>22</sup> Belda, S., et al., 'The Short-Term Prediction of Length of Day Using 1D Convolutional Neural Networks (1D CNN),' *Sensors*, 2022.

<sup>23</sup> Abdeljaber, O., et al., '1D Convolutional Neural Networks and Applications: a Survey,' *Mechanical Systems and Signal Processing*, 2021.

<sup>24</sup> Ibid.

<sup>25</sup> Ibid.

<sup>26</sup> Abdeljaber, O., et al., 'Operating Machine Learning Across Natural Language Processing Techniques for Improvement of Fabricating News Models' *International Journal of Science System Research*, 2020.

<sup>27</sup> Ibid.

<sup>28</sup> Ibid.

for the  $i - th$  data point (the real, unseen by the model, observation), and  $f(x_i)$  the predicted value produced by the model, for the  $i - th$  data point.

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - f(x_i))^2$$

**Equation 6: Mean squared error loss**

#### 2.2.2.4 Transformers

Another, more novel, ML model for studying sequential data, transformers can be best characterised by the addition of self-attention mechanisms, often found in NLP. Through self-attention mechanisms, the model focuses on different parts of the input sequence when making estimates for a sequence of  $N$  elements, denoted  $X = [x_1, x_2, \dots, x_N]$ . These mechanisms then compute a new sequence, often referred to as the contextual or weighted sequence, denoted as  $Z = [z_1, z_2, \dots, z_N]$ . For each variable it is provided, three sets of vectors are computed: (1) query vectors, (2) key vectors, and (3) value vectors. The first represents a given variable's importance, what the model needs to pay attention to, the second, how much other variables will affect the given one, and the third, represents the content of said variable.

These vectors are computed as linear transformations of the input sequence  $X$  using the learned weight matrices. For a specific variable,  $x_i$ , the query, key, and value vectors can be computed as shown below.<sup>29</sup>

$$q_i = W_q \cdot x_i(\text{Query}) \qquad k_i = W_k \cdot x_i(\text{Key}) \qquad v_i = W_v \cdot x_i(\text{Value})$$

**Equation 7:**

**Equation 8:**

**Equation 9:**

**Computation of query vector**

**Computation of key vector**

**Computation of value vector**

$W_q$ ,  $W_k$ , and  $W_v$  are the learned weight matrices for a given variable to the self-attention mechanism. The self-attention mechanism computes attention weights for each variable in the input sequence, computed through similarity function. Often the dot or scale dot product:

$$Attention(q_i, k_j) = \frac{q_i \cdot k_j}{\sqrt{d_k}}$$

**Equation 10: Computation of self-attention mechanism<sup>30</sup>**

Where  $d_k$  is the key vectors' dimensions. After calculating the attention weights, the mechanism generates the weighted sum of the value vectors to obtain an output for each element. This weighted sum incorporates information from all the variables in the input sequence, where the importance of each is determined by the attention weights. Thus, the self-attention mechanism allows the model to focus on different parts of the input sequence when making an estimate, highlighting it as a powerful tool for capturing long-range dependencies and context in sequences.<sup>31</sup>

$$z_i = \sum_{j=1}^N Attention(q_i, k_j) \cdot v_j$$

**Equation 11: Computation of the weighted sum of value vectors**

Our architecture combines convolutional layers for feature extraction with transformer layers for capturing temporal dependencies in time series data. Transformer layers are stacked to enhance feature representation. Dropout and layer normalisation improve model robustness, followed by dense layers leading to the output layer. The model is trained to minimise mean squared error loss and use mean absolute error for evaluation. This design exhibits the use of transformers to produce precise estimates by capturing nuanced temporal patterns.

<sup>29</sup> Ibid.

<sup>30</sup> Ibid.

<sup>31</sup> Ibid.

### 2.2.2.5 Ensemble Learning

To leverage the strengths of both the 1-dimensional CNN and the transformers' architecture, an ensemble method was applied. An ensemble takes advantage of the strengths and capacity possessed by each independent architecture,<sup>32</sup> using either a voting between or averaging of outputs to produce a final output. For the Dynamic Flows Modeler, averaging was chosen as the ideal ensemble method. The average function takes the element-wise average of the estimates produced by the models. The ensemble model is then trained to minimise the mean squared error loss, and mean absolute error is used for evaluation, as is the case for the transformers model.

$$\text{ensembleEstimate} = \text{average}(\text{CNNEstimate}, \text{TransformersEstimate})$$

#### Equation 12: Dynamic Flows Modeler ensemble averaging function

## 2.3 Current Advancement and Demo

The Dynamic Flows Modeler's current state of advancement meets the GA's expectation with respect to delivering an AI-based tool which could understand the "spatiotemporal evolution of D&FN induced and D&FN-enabled of offline crimes"<sup>33</sup> and, from such understanding, can provide LEAs with significant, increased capacity in "the identification and deployment of relevant security measures related to D&FN".<sup>34</sup> The estimates signal the direction of crime and, therefore, even with a margin of error, provide end-user LEAs with an understanding of what types of criminal behaviour are liable to be most impacted by D&FN. As such, they can adjust to these changes, reacting to them with their expertise knowledge of necessary counter measures and priorities. Moreover, the impact of each crime type is provided downstream in the FERMI platform, with the Dynamic Flows Modeler's output being provided to the Behaviour Profiler & Socio-economic Analyser.

That being said, the GA commitment to "produc[e] bigdata-based profiling of authors and victims of D&FN induced and D&FN-enabled"<sup>35</sup> and combine this with event-time-victim-author relations is, at the current stage, beyond the ability of the Dynamic Flows Modeler. Attempts are still being made to find big-data on crime incidents, which provides information on the individual or entity victimised by a crime or information on offenders. Whilst data-scarcity on past crime events was overcome by T3.4 on swarm learning and switching to an American data context to train the model, these approaches have not proven to be solutions with respect to victim/author commitments. Currently, arrest records, rather than crime incidents, are being investigated to determine if they can provide a satisfactory number of observations. Alternatively, the limited American municipalities with victim/author data may be used to investigate event-time-victim-author relations, generally, providing FERMI with the information necessary to meet our GA commitments.

**With respect to the demonstration of the FERMI platform, the Dynamic Flows Modeler is ready to be integrated.** That being said, development of the Dynamic Flows Modeler is continuous, with new approaches and alterations proposed and trailed nearly daily. With that in mind, the technology is continually improving and, therefore, the later the date of integration the better performing the Dynamic Flows Modeler will be.

## 2.4 Next Steps

The proceeding activities, in developing the Dynamic Flows Modeler, will be focused on three tasks: (1) extracting left-wing extremism from the NELA-GT datasets, (2) locating crime incident data with victim-author information, and (3) conducting thorough analysis on the author-victim relationships. The subsequent sub-sections will briefly explain each of these next steps and which GA commitments they help meet.

<sup>32</sup> Murali, V., 'Everything you Need to Know about Ensemble Learning,' *Medium*, 2021; Atiya, A. F., 'Why does Forecast Combination Work So Well?' *International Journal of Forecasting*, 2020.

<sup>33</sup> 'Grant Agreement: Project 101073980 – FERMI – HORIZON-CL3-2021-FCT-01,' European Research Executive Agency, 2021.

<sup>34</sup> Ibid.

<sup>35</sup> Ibid.

---

### **2.4.1 Left-wing Extremism**

Extracting left-wing extremist content is a necessary pre-processing task, in order to train the Dynamic Flows Modeler with the intensity of said D&FN topic, and, in turn, meet GA commitments to study the topic's impact on offline crime. However, as explained earlier, it proved too costly, in terms of time and resources, to complete now. Despite following a similar process as was used to extract right-wing extremism the obtained sample size was relatively small and not entirely suitable for comprehensive analysis. There is action already in place to solve this issue: manual classification of articles from the NELA-GT dataset, an approach that requires a rather long period of time to be successful. Thus, while this is underway, we are also looking for other datasets of D&FN that contain a larger sample of left-wing content than NELA-GT. X's application program interface is one of these alternative datasets, though, the recent changes to access permission as required by the newly passed Digital Services Act remain to be fully examined.

### **2.4.2 Victim-Author Data and Relationship**

Out of the 31 American places that published, publicly, crime incident data, only one (Cincinnati, Ohio) provided observations on victims and authors that were of a quality and quantity suited for analysis. Unfortunately, the same absence of victim-author data was presented in the crime data provided by partner LEA. This issue is endemic to incident-level crime data as victims are often entitled to a high degree of privacy and prosecution of offenders may take several years, meaning they are also maintaining a right to privacy, if they are even known when a crime event is filed by LEAs. Alternative datasets, such as arrest and conviction records, are being reviewed and attempts to further cooperation with partner LEAs will be attempted, while being sure not to overstep and violate the privacy of the accused or the victim, for a given offline crime event.

### **2.4.3 Retraining at Regular Intervals**

The Dynamics Flows Modeler has been trained, as previously mentioned, on data from 2020 – 2022. Considering the General Project Review Consolidated Report's recommendation to "focus on constantly updated data sets,"<sup>36</sup> however, as new D&FN, socio-economic, and crime data become available, the models can be retrained to ensure the persistent accuracy and the appreciation in any changes in the nexus between online D&FN and offline criminal behaviour. This retraining can be performed once the necessary data has become available or when an eventual end-user LEA provides their private, attune data via the swarm learning infrastructure (see section 4).

---

<sup>36</sup> General Project Review Consolidated Report, p.2.

### 3 Task 3.2 – The Spread Analyser

The Spread Analyser is another fundamental component of the FERMI platform, with a crucial role in the investigation process. The aim is to **provide end-users with a graphical representation of the provided D&FN’s diffusion online**, specifically in social media platforms. Said graphical representation is analysed and infused with additional information to offer improved value to the user, beyond a mere visualisation. Through data extractors, data analysis, graph analysis, and ML, each social media post relevant to the given D&FN’s diffusion is accompanied with information to support the end-users understanding of the network they are investigating. Importantly, among this provided information, is **the identification of bots, if any are present, behind the posts and a measure for each posts’ influence**. As such, **the end-user can gain a comprehensive understanding of the D&FN in question’s spread**. Moreover, the output of the Spread Analyser is a crucial building block in other FERMI technologies’ analysis.

The Spread Analyser complies with legal and ethics constraints as laid out in the WP7 deliverables, including the human-in-the-loop approach. Amongst other things, the Spread Analyser does not broadly retrieve social media posts but is being fed with them by the end-users. They are the ones who decide in accordance with the legal framework they are bound by whether a post’s spread, influence and human vs. bot origin needs to be analysed, which can greatly advance evidence-gathering. The selection of social media posts for the pilots is informed by the FERMI project’s above-mentioned guiding definition of disinformation, including the following building blocks (1) factual or misleading nature of the information; (2) intention of the actors to spread such information they know to be false to obtain economic gain or deceive the public and (3) public harm.

The rest of section 3 is structured as follows, 3.1 focuses on the practical description of the component and its implementation; 3.2 provides a thorough methodological and technical brief of what has been accomplished and, importantly, its compliance with the GA; subsection 3.3 focuses on the current state of advancement and implementation, and subsection 3.4 addresses the next steps in development and their envisioned outcomes.

#### 3.1 Practical Description

To achieve its desired, and GA committed to, objectives, **the Spread Analyser relies on variety of services**. First among these objectives is the production of a social media post graph, representing the spread of the D&FN provided by the platform’s end-user. As stated in the GA “creating a tool that will take as input news already classified as disinformation and will be able to trace and map this news to their main actors/accounts which are responsible for creating and spreading the disinformation across the network.”<sup>37</sup> Supporting insights, such as the ability to “classify these accounts as physical persons or bots and” the requirement to “offer for every account an influence index in order to understand their power over the network,”<sup>38</sup> are also clearly laid out in the GA. This deliverable focuses on the first version of the component, with the primary focus of describing the progress towards the above-mentioned objectives/GA requirements and the following paragraphs are dedicated to describing the different intracomponent systems created in said progress.

##### 3.1.1 Design

The Spread Analyser consists of different services to provide the user with substantial information and insights into each investigated social media post. As a result, **the component is divided into distinct services, functioning as intracomponent system elements**. The graph builder and Social Media Application Programming Interface (API) Crawler intracomponent systems are responsible for extracting data from a social media platform and building the investigation graph. As in the GA, these intracomponent systems analyse “the potential spread of the D&FN”,<sup>39</sup> and, more specifically, trace and map this news to their main actors/accounts

<sup>37</sup> Ibid.

<sup>38</sup> Ibid.

<sup>39</sup> Unlike the what the GA states in the “Description of individual components and offerings” this analysis is not “driven by the analyses carried out within the Dynamic Flows modeler.” The contributions of both tools are aligned to support the platform and provide LEAs with the capabilities to collect evidence and to assess the resultant threat in terms of

which are responsible for creating and spreading the disinformation across the network.”<sup>40</sup> Subsequently, **the built intracomponent system elements, again in accordance with the GA**, enrich the identified spread network with an influence index, “in order to understand [an identified users] power over the network,”<sup>41</sup> and “classify these accounts as physical persons or bots.”<sup>42</sup> In accomplishing the latter objective, two smaller objectives were identified: the development of (1) an insight extractor, to enrich the graph, and (2) graph analyser. The insight extractor has access to a ML-based model and a graph analyser to produce insights on the nodes (comprised of social media posts) of the investigation’s graph. The ML model is a classifier that, leveraging social media post and public user data, produces a classification of the social media post’s author, accompanied by a confidence index. The graph analyser is an influence assessment estimator that assigns to each node an influence score based on its relationship with the rest of the nodes in the network. Effectively managing the different services implemented required an orchestrator intracomponent system to be developed. Figure 7 illustrates the intracomponent systems’ architecture, in the following subsections, each intracomponent systems’ functionality will be presented.

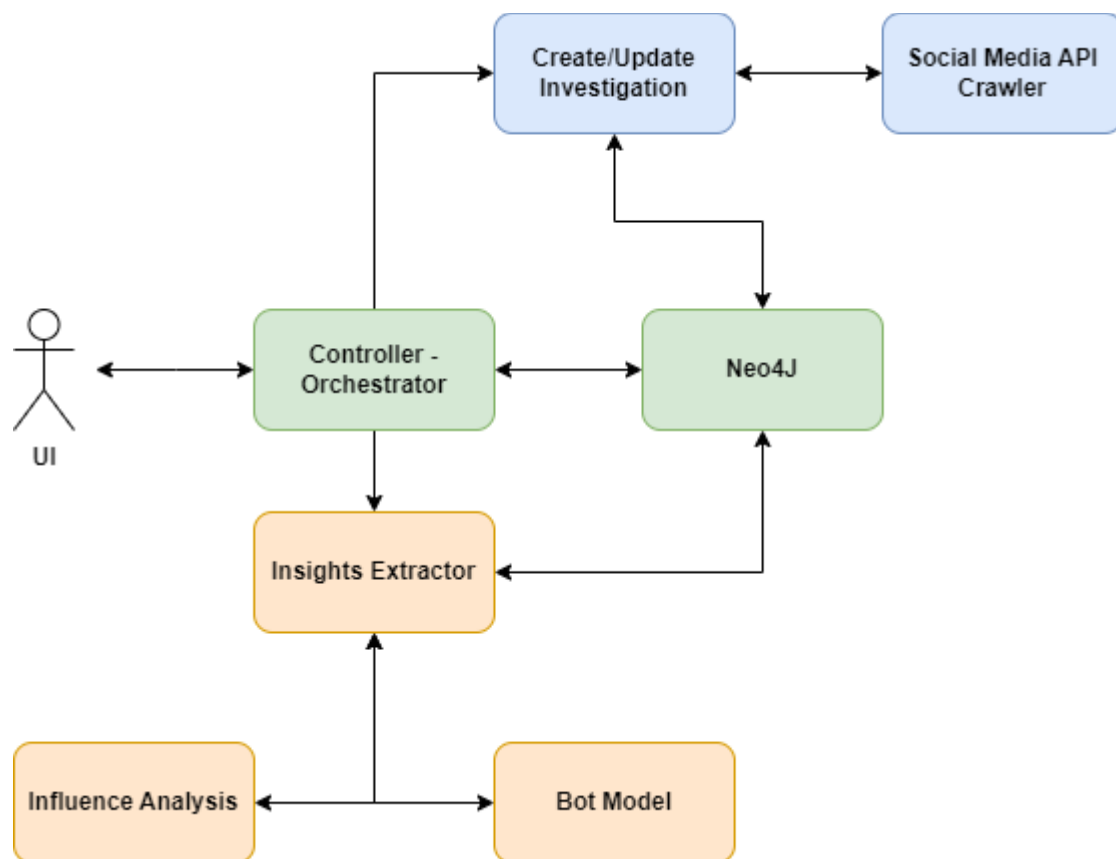


Figure 7: Spread Analyser’s intracomponent systems’ architecture

### 3.1.1.1 Network Graph

The graph building intracomponent system is a service **responsible for developing the network of social media posts related to the investigation’s starting D&FN post**. The structure of the graph consists of

changes in the crime landscape that are addressed by the Dynamic Flows Modeler. The subsequent remark, that the spread analyser is aimed at “analysing the potential impact of these spreads, in a socioeconomic framework,” is covered by the Socioeconomic Analyser, as explained, with brevity, in section 6, subsection 6.1. For a more thorough account of the Socioeconomic Analyser, see Deliverable 3.3.

<sup>40</sup> ‘Grant Agreement: Project 101073980 – FERMI – HORIZON-CL3-2021-FCT-01,’ *European Research Executive Agency*, 2021.

<sup>41</sup> Ibid.

<sup>42</sup> Ibid.

nodes and edges which represent relationships between different social media posts. The graph's structure is layered, to ensure that each layer adds depth, by expanding the information provided to the end-user as needed. The initial D&FN post is placed at the centre of the graph and is considered as layer zero. Each additional layer represents social media posts related to a social media post from the previous layer, demonstrating the branching between different social media posts. While this process requires implementing a structured building policy, a range of different limitations present between social media platform's varying APIs have to be considered to effectively plan graph building. Said limitations include the accessible depth of historical data,<sup>43</sup> the number of requests per minute<sup>44</sup> and the total requests allowed per month.<sup>45</sup>

Consequently, a technique implemented, achieved significant minimisation of the required number of API calls per investigation. This was accomplished by finding optimal calls that leverage relationships between social media posts to simultaneously retrieve data from multiple posts with only using a single API call. Additionally, policies on graph expansion, introduced a prioritised expansion of nodes and offered a balanced approach to node expansion and reached investigation depth. Furthermore, for the duration of graph building **the developed graph is being stored in a Neo4J database** instance managed by the graph building service.

The graph building intracomponent can build network graphs from a wide variety of social media platform given that relationships between posts are available. Social media platforms do not always have the same type of relationships or classify the same type of relationship with different names (e.g., reply in X is the same as comment in Facebook). As a result, development to adapt per social media platform is required but the main graph building process is social media platform agnostic.

### 3.1.1.2 Social Media API Crawler

The development of the graph is supported by the Social Media API Crawler intracomponent system. This service has two goals: the first is the enrichment of the developed graph by extracting information, regarding the social media posts and their authors, through social media's APIs for each node of the network. The second is the management of the API calls to ensure the Social Media API Crawler is in alignment with principles of purpose limitation and data minimisation. Wherein, any tool processing personal data should ensure that data collection is minimised to only the data necessary to pursue the objective of said data's processing.<sup>46</sup> This means that the Social Media API Crawler's API calls, in building the graph are limited, to respect the personal data processing norms, even if the data is public, as in the case of most social media platforms' posts. Ensuring the management of the API calls is a multifactorial problem, since the various restrictions (from social media platforms) imposed by API call-type must be appreciated to not exceed the total monthly API calls limit, while simultaneously being able to have API calls available for more than one investigation. All the while ensuring a minimised number of calls to pursue the objective at hand.

As stated above, each social media platform enforces restrictions and limitations affecting the access to information both in volume and available data. Additionally, each social media platform offers different API solutions with distinct capabilities. Consequently, both goals of the Social Media API Crawler are significantly affected per social media platform. The component need merely be adapted per social media platform to suit to the APIs provided, considering the access intricacies and limitations for each social media platform.

### 3.1.1.3 Insights Extractor

The insights extractor is an intracomponent system that functions as a handler for the services and models enhancing the investigation graph. Additionally, it consumes and makes API requests, within the component, to receive graphs and share their updated versions following the investigation enrichment services application.

<sup>43</sup> 'Search Tweets,' *X Developer Platform*, n.d.

<sup>44</sup> 'Quote Tweets,' *X Developer Platform*, n.d.

<sup>45</sup> 'Getting Started,' *X Developer Platform*, n.d.

<sup>46</sup> OJ L, 2016/679, 4.5.2016, ELI: <https://data.europa.eu/eli/reg/2016/679/oj>



### 3.1.1.4 Influence Analyser

The influence analyser is charged with the task of analysing the graph that has been created from the initial D&FN input. The **influence analyser uses the Graph Data Science (GDS) Library of the Neo4J in order to rank the nodes of the graph from the most influential to the least influential one**. By doing this it also meets the GA requirement to ensure that “for graph data, graph clustering and graph machine learning algorithms will be developed to detect highly influential nodes spreading misinformation [this is the wording used by the GA, albeit the FERMI consortium has agreed to use a guiding definition of disinformation (as opposed to misinformation) to guide its analysis of D&FN, as explained above and as laid out in greater detail in D2.1]: the Neo4J libraries for graph Data Science will be used and expanded through new efficient algorithms for ‘centrality’ calculations on the overlaid graphs emerging from misinformation spreading.”<sup>47</sup> Specifically, **for the calculation of the most influential nodes, centrality algorithms have been used along with the PageRank algorithm, from the Neo4J GDS Library**. These algorithms take advantage of the connections between the nodes and quickly find the nodes that are the most important ones among the others. The influence analyser module is able to perform its task regardless of the social media platform that is used. The sole requirement that exists, is for the graph to have been created from the graph creator component, which indicates how flexible the influence analyser module is.

### 3.1.1.5 Bot Model

The bot model has the task of discerning if a particular node that exists in the graph is representing an actual human user or is a bot account. For the creation of the system, deep learning techniques were deployed and, specifically, an artificial neural network was developed, whose task is to detect which target class (bot or human) each node belongs to. Furthermore, the use of deep learning techniques satisfies the GA, which states “the tool will be able to classify these accounts as physical persons or bots and it will offer for every account an influence index in order to understand their power over the network. For datasets containing ground truth labels,<sup>48</sup> advanced deep learning techniques tailored to NLP will be employed, in particular the attention mechanism will be combined with recurrent deep networks.”<sup>49</sup> The model performs this by taking into account certain metadata which are created when the graph is created. In addition, the model, firstly, is trained with an open-source labelled dataset, due to the fact that the data gathered for the creation of the graph do not include the information required for the bot or human classification. Then, the model is applied to the graph data and updates the corresponding bot or human field in the graph dataset. The bot model, as mentioned, has been trained using open-source dataset from the X-platform. Considering that, upon integration of additional social media platforms, the developed deep learning model may be retrained with data sourced other social media platforms, as per end-user needs accounting for variations in fields’ names and availabilities.

<sup>47</sup> ‘Grant Agreement: Project 101073980 – FERMI – HORIZON-CL3-2021-FCT-01,’ European Research Executive Agency, 2021.

<sup>48</sup> Considering the use of ground truth labels, the following remark of the GA has been overtaken by events: “for datasets without ground truth to train supervised classifiers, specialized algorithms for K-Means clustering that offer optimal or near-optimal solutions in the large K-value domain will be employed to detect small clusters spreading disinformation” (‘Grant Agreement: Project 101073980 – FERMI – HORIZON-CL3-2021-FCT-01,’ *European Research Executive Agency*, 2021). The fall-back option of using non-ground truth labels simply is no longer necessary, as labelled datasets (ground truth) are available.

<sup>49</sup> In particular, the GA alludes to GRU, LSTM, etc; elsewhere, KNN, K-Means, as well as “more advance[d] DL models such as MLPs, CNNs, [and] RNNs” are invoked (‘Grant Agreement: Project 101073980 – FERMI – HORIZON-CL3-2021-FCT-01,’ *European Research Executive Agency*, 2021). All these techniques have been analysed in-depth to assess whether they might make a valuable contribution to the tool at stake. It has been concluded that for the task of the classification of accounts, as bots or humans, based on the data collected from the X API neural networks and, specifically, MLP seem to be sufficient deep learning techniques to tackle the problem and the use of GRUs or LSTMs (types of RNN) do not seem a relevant solution, based on the data at hand. In addition, it was possible to find similar open-source data to the data gathered from the X API, that also contain ground truth data to train our MLP model. Thus, it was not necessary to use KNN or K-means algorithms which were described only for the case that data without ground truth would be available.

### 3.1.1.6 Orchestrator

The orchestrator or controller intracomponent system is a service supporting the rest of the intracomponent systems implementing the investigation objectives. **This service maintains a queue of incoming investigation requests by the user to better handle the intracomponent services activated per request and ensure that all requests will be served.** Furthermore, the service manages both intracomponent communications between services and external communications with the FERMI platform-dependent components. Both internal and external communications are implemented with representational state transfer API calls, using the version 3 OpenAPI specification to ensure uniformity and industry standards.<sup>50</sup>

## 3.2 Technical Description

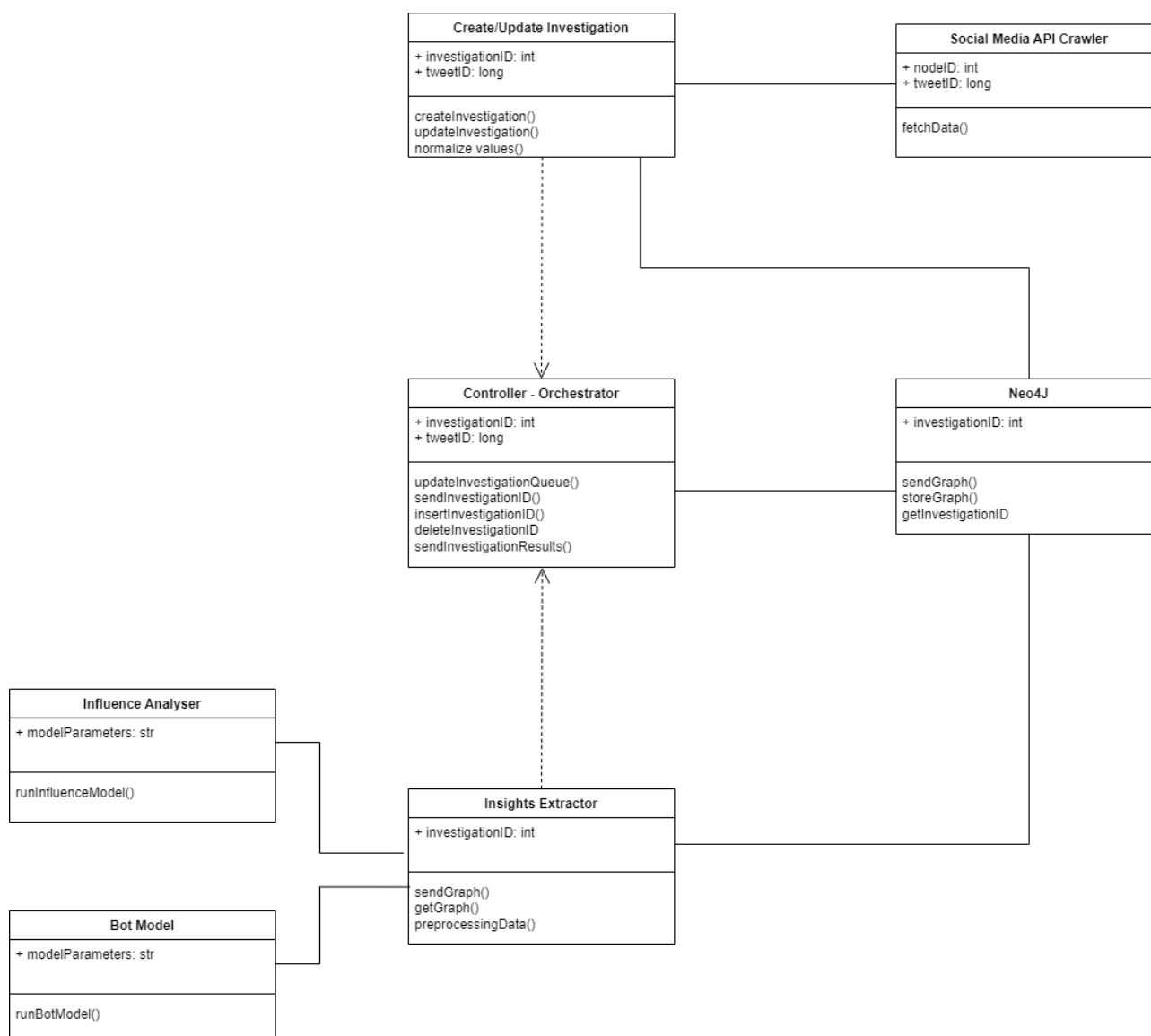
In 3.2, we will describe the intricate, inner workings and methodology of the module, expanding on the practical description provided previously. The module is characterised by its multi-objective nature, addressing several critical facets of D&FN analysis. Those can be identified in building the D&FN graph, encapsulating data directly from social media platforms, performing the spread analysis, which determines, accurately, the crucially influential actor of the mapped network, and, through applying the bot classifier, estimates if the investigated D&FN post was produced by a human or a machine.

### 3.2.1 Design

The functionalities described above are implemented inside our five intracomponent systems. Starting with the Social Media API Crawler, used to retrieve social media data; the graph builder, which creates the D&FN graph; the neo4j database, a space in which the graph is not only stored but also visualised; the insights extractor, that applies our ML models and analysis services to the graph, and the orchestrator which synchronises the above-mentioned modules and handles the communication with the rest of the FERMI platform. Figure 8 accurately depicts the class diagram of our solution. In the following paragraphs we will elaborate on each component of the Spread Analyser, individually, and how it fits with our approach.

---

<sup>50</sup> Haupt, F., 'A Model-Driven Approach for REST Compliant Services,' *2014 IEEE International Conference on Web Services*, 2014.



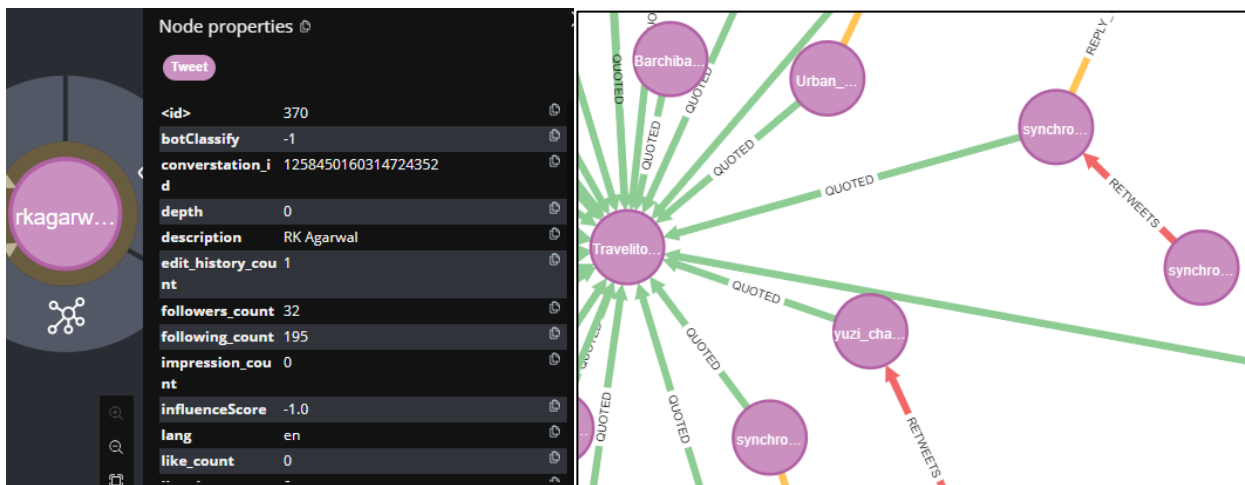
**Figure 8: Spread Analyser class diagram**

### 3.2.1.1 Network Graph

The importance of the network graph is best seen when one analyses the kind of information that it depicts, as well as the process followed while constructing it. The starting point is a social media post;<sup>51</sup> although every post, as an entity, includes a wide variety of information that defines it (name of their creator, number of likes and reposts, its content, etc.), said information, alone, is not enough to construct a high value graph. At least not one that provides the amount and quality of information required for a successful investigation. To tackle this obstacle, **FERMI searches further, examining the posts’ authors. Information on the authors’ location, popularity, and the details regarding their overall presence on the platform.** Each node of the graph encapsulates all this information in a single point providing easy and fast access to it.

<sup>51</sup> As explained above, the LEA end-users will feed the FERMI platform, including the Spread Analyser with such posts. Accordingly, the FERMI platform will not broadly collect social media posts but depend on end-user input that can and will be provided in strict compliance with the relevant LEAs’ legal limitations and in accordance with the FERMI project’s guiding definition of disinformation. Against this backdrop, the to-be-analysed social media posts will include remarks that require LEA action and, if possible, 1) factual or misleading nature of the information; 2) intention of the actors to spread such information they know to be false to obtain economic gain or deceive the public; 3) public harm.

Nodes' connections (graph edges) represent the relation type between two posts and this relation can differ between “retweets”, “reply to” and “quoted”. In figures 9 and 10 we can see a portion of the structured information inside a node and the relationship between nodes.



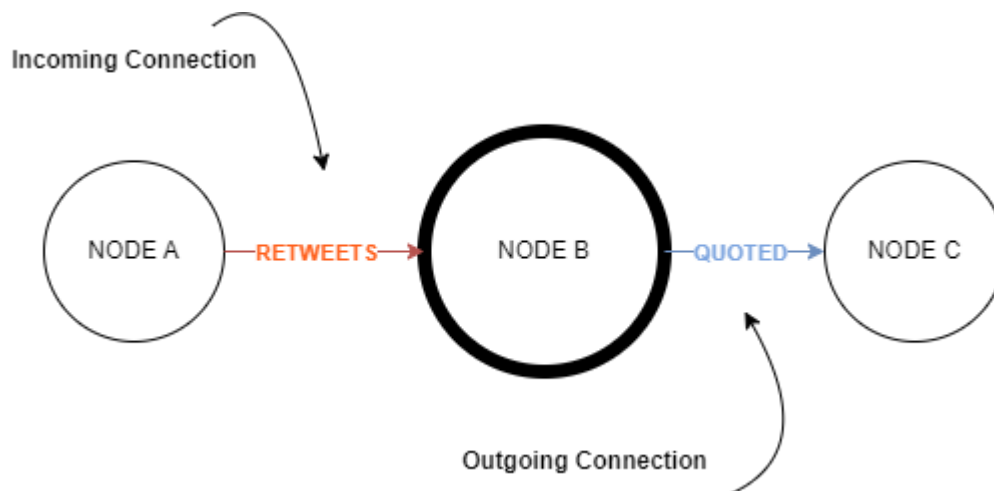
**Figure 9 (left) & Figure 10 (right): Graph nodes' information and relationships**

In this step, a significant challenge was faced. In order to acquire the information required to construct a node it is mandatory to make calls using a social media's API. With many APIs accessible only through paid subscriptions prior to the Digital Services Act entering into force, the consortium chose to move ahead with X for development and validation of the technologies, particularly those in T3.2. The license acquired limited API calls to 5 per 15 minutes, as well as the monthly limit on total calls. The time of building the graph is highly dependent on the amount of information we can gather during a period of time.

When constructing a network, having as a starting point a single social media post, two possible outcomes may arise. The first occurs when the investigated post has not reached high popularity on the social media platform, which, ultimately, means that only a handful of users have interacted with it. This scenario makes the graph building process relatively fast and creates a graph representing a small cluster of users and posts. In this case, the number of requests per specific unit of time API limitations of social media platforms do not pose a major issue, as the prolongation of the building process is minimal or, even, non-existent. The latter occurs when the post has increased popularity and multiple users engage with it. In that case, the network becomes complex, and the process required to build the graph requires more time, making it impossible to map all of the network and investigate every node, given the limitations introduced in social media APIs.

In order to overcome this obstacle, **we have advanced the graph building methodology, aiming to minimise API calls and save time, while ensuring the maximum expansion of the network.** To achieve this, the first step was to tag each node with the custom property of depth. Depth represents the distance - count of edges - between the current node with the initial node of the investigation (the initial node describes the post that started the investigation). **This creates the effect of different nodes sharing the same property and value pair, meaning that two or more nodes can have the same depth.** When grouping nodes with the same depth level, we can form different layers inside our graph, this means that a layer level consists of all the nodes with depth equal to the layer level. Layer 0 only includes the starting investigation point (depth 0), the investigation social media post provided by the user. These layers play a fundamental role in the strategy we use to expand our graph.

Another crucial aspect is the direction in which each node can be expanded to. **A node can form both incoming and outgoing connections.** Incoming connections form with other nodes that retweet, reply or quote our node, whereas outgoing connections form with the nodes that are retweeted, replied or quoted by our node. For example, the following figure explains the connection type when inspecting NODE B:

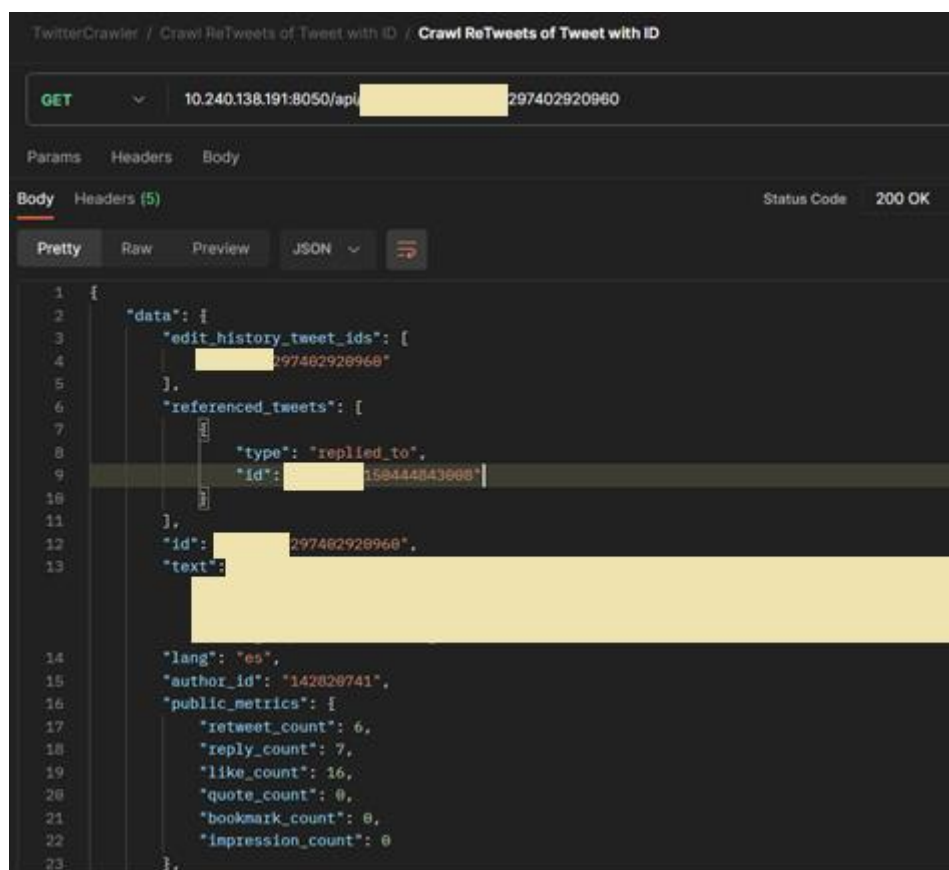


**Figure 11: Example of node relationship type connections**

Combining these two factors we can form the expansion strategy. As mentioned above, to save API calls, not every node available can be investigated. The expansion takes place from a lower depth level layer to a higher depth level layer. The first step is to select which nodes will be expanded and analysed, which is accomplished by creating a priority queue and assigning a priority value to the available nodes of the layer. **Only nodes with higher priority are incorporated into the investigation and expanded.** The outcome of this process is the next layer consisting of the expansion of the selected nodes. When expanding, it is crucial to ensure the inclusion of nodes from both directions, validating the continuity inside the graph and also ensure that optimal nodes are selected. Optimal nodes are defined as those with a higher probability of revealing a bigger network behind them.

### 3.2.1.2 Data Collection

When gathering data for an investigation it is crucial to ensure the enforcement of social media platform's API's limits. To that end we have created a component responsible of keeping track of the amount of the requests sent, time elapsed between them, and calculating the waiting time in order not to overcome our limit. It also applies pre-defined parameters which enrich the social media platform's response and help us gather all the information available. In order to achieve this, we have grouped the distinct type of requests to retweets, quotes, replied and retweeted by. This was crucial, as specific social media platform endpoints only provide information created during the last 7 days. Figure 12 provides an example of a social media platform API call and its response.



```

1  {
2    "data": {
3      "edit_history_tweet_ids": [
4        "[redacted]297402920960"
5      ],
6      "referenced_tweets": [
7        {
8          "type": "replied_to",
9          "id": "[redacted]158444843888"
10         },
11      ],
12      "id": "[redacted]297402920960",
13      "text": "[redacted]",
14      "lang": "es",
15      "author_id": "142826741",
16      "public_metrics": {
17        "retweet_count": 6,
18        "reply_count": 7,
19        "like_count": 16,
20        "quote_count": 0,
21        "bookmark_count": 0,
22        "impression_count": 0
23      }
24    }
25  }

```

**Figure 12: Example of X API-call response**

### 3.2.1.3 Insights Extractor

The insights extractor is an intracomponent system supporting the graph enrichment services of the source analyser component. Its functionality begins with an API consumer that receives a graph from the component to further supplement it. This is possible by employing the influence analyser and bot classifier model. After receiving the updated graph with the input from both services, it shares it with the component for the next action in the intracomponent pipeline handled by the orchestrator/controller service system.

### 3.2.1.4 Influence Analyser

The influence analyser service is responsible for analysing the graph and finding the most influential nodes. It was developed through Neo4J's Graph Data Science library, as denoted in the task's GA description. "Machine learning algorithms will be developed to detect highly influential nodes spreading misinformation [as clarified above, this is the wording used by the GA, albeit the FERMI consortium has agreed to use a guiding definition of disinformation (as opposed to misinformation) to guide its analysis of D&FN]: the Neo4J libraries for graph Data Science will be used and expanded through new efficient algorithms for 'centrality' calculations on the overlaid graphs emerging from misinformation spreading." Betweenness centrality<sup>52</sup> and PageRank algorithms were developed, however, PageRank algorithms provided greater accuracy and had the more logically consistent results. In turn, PageRank algorithms were chosen for the first version of the influence analyser. The algorithm measures the importance of each node within the graph based on the number of incoming relationships and their importance which can be considered or not during the analysis depending on the actual examination process. In case further fine tuning will be needed, different weights for the relationships can be incorporated; the mathematic formula behind the PageRank is reported in Equation 12.

<sup>52</sup> Freeman, L. C., 'A Set of Measures of Centrality Based on Betweenness,' *Sociometry*, 1977.

$$PR(a) = \frac{(1 - d)}{N} + d \sum_{x \in N(a)} \frac{PR(x)}{C(x)}$$

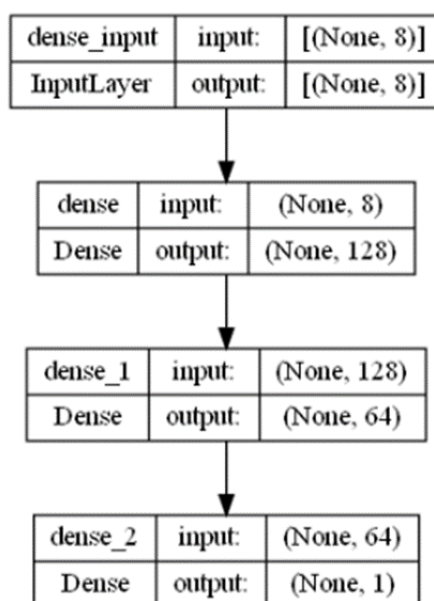
### Equation 12: PageRank formula

Where  $N$  is the number of nodes,  $N(x)$  denotes the set of neighbouring nodes with links to node  $a$ ,  $C(x)$  is the number of outgoing links in node  $a$  and  $d$  is the damping factor. The contribution of  $PR(x)$  from a neighbouring page  $x$  is divided by  $C(x)$  assuming each link has an equal chance to be selected. The damping factor  $d$  can be set to any value between 0 (inclusive) and 1 (exclusive) but is usually set to 0.85. PageRank is an iterative algorithm which means that it is run iteratively until it updates a candidate solution until convergence. Furthermore, it is important to state that the initial user that has created the post gets the highest PageRank score, with the further nodes receiving lower scores. Overall, PageRank appears to be a very suitable algorithm to find nodes with the most influence within the graph network. Finally, the module and thus the algorithm can be applied on any built graph in the Neo4j platform without any limitations based on the social media platform from which the graph was built on.

#### 3.2.1.5 Bot Model

The bot classification model serves to identify if a specific node within the created graph network was created by a human or by an artificial entity, that is, a bot. Deep learning technique were used and, specifically, an artificial neural network was developed to accomplish the goal. Due to the fact that the data gathered to create the graph network do not specify whether a certain node is a bot or not, an open source, labelled with the information of bot or human, dataset was used in training the model. Then, to apply the model to the graph's data, both datasets were harmonised to contain the same features. Subsequently, the model was able to be applied to both. The main features that constitute the dataset is the description, followers (of the user), user's following (i.e., the accounts the user follows), geolocation (if available), language, location given in the description, average social media posts made per day by the user, how many days the account has been active and account type (bot or human), which acts as the target feature. As mentioned, data from the X-platform were used so the model could be trained and developed. This does not limit the usage of the module only in a specific platform but given the existence of quality data from different social media platforms the model can be retrained, remaining effective and able to detect bots based on data and graphs built on different social media platforms. The features were pre-processed so they can be used as input to the neural network. The features which had text as values, for the first version of the model, were handled as binary variables, meaning that new variables were created containing the value 1 in the case that the observation existed and 0 in case the observation did not exist. The features which contained continuous numbers as values were normalised to ensure that the neural network was able to better handle these values and to extract meaningful patterns from

them. For the model development, a multilayer perceptron<sup>53</sup> (MLP) was created with two hidden layers of 128 and 64 neurons, respectively, as presented in the figure below.



**Figure 13: Bot classifier's architecture**

The number of hidden layers and neurons were chosen after having evaluated the results of the model on the validation set, during training, and aiming to have the model as lightweight (in terms of speed) as possible. The developed model is capable of achieving an accuracy of 83% and a weighted F1-Score of 82%.

	precision	recall	f1-score	support
bot	0.78	0.67	0.72	2454
human	0.85	0.91	0.88	4938
accuracy			0.83	7392
macro avg	0.81	0.79	0.80	7392
weighted avg	0.82	0.83	0.82	7392

**Figure 14: Classification report**

The model is better at forecasting when the user is a human than it is at forecasting a bot user. This behaviour is partially due to the fact that, in the data, bot accounts are far less frequent. However, the first version of this model is able to achieve decent performance given the fact that it is able to do the basic classification task that it was created for. In the next phases of the project, the team will work to improve the F1-Score and the model's accuracy even further.

### 3.2.1.6 Orchestrator

The orchestrator component system supports and controls the activation of the rest of the intracomponent systems while also being responsible for all external communication with the platform components. The orchestrator is, importantly, the creator of new investigations, based on a call from the platform. This is achieved through consuming API calls requesting new investigations to be initiated from the platform. While this is a straightforward function, the component must be able to handle multiple requests, simultaneously, to avoid missing investigation requests and, at the same time, assist the control and activation of different services in the component. To that end, a queue service has been established with a dual purpose: (1) initially to store investigation requests and avoid duplication of requests and (2) to handle requests

<sup>53</sup> Balas, V.E., et al., 'Multilayer perceptron and neural networks,' *WSEAS Transactions on Circuits and Systems*, 2009.



depending on the availability of the rest of the intracomponent systems. That way, the services of the component can be used optimally and several investigations can be performed in different stages of analysis.

The second service of the component is the control of the intracomponent systems. This service relies on the implemented queue service to get the investigation request data and transmit them via API calls to the intracomponent systems for analysis and processing. In particular, it handles the investigation creation pipeline, initially activating and temporarily storing the results from the graph building system component, followed by the activation and output handling of the insights extractor component system. After the investigation is completed, the results are sent through API calls to the next platform components needing them for their analysis.

### 3.3 Current Advancement and Demo

At this moment in production, a fully designed Spread Analyser has been achieved. All the functions that constitute the completed system are described in detail, from the initial function, which crawls social media platform data; the function which creates the investigation, along with the creation of the graph in the Neo4J tool; the controllers of said functions, and, finally, the influence analyser together with the bot detection model. Additionally, the graph creator is fully functional and ready to be deployed for the means of the platform. It can use a single social media post as input and initiate the various processes to fetch all the posts connected with this single social media post and accordingly to create the graph through the Neo4J tool as stated in the GA.

The Spread Analyser is capable of tracing and mapping the spread of a user-provided D&FN on social media platforms back to its authors. Furthermore, the influence analyser, based on the PageRank algorithm, produces, for every social media account in the network, an influence index, establishing their power over the network as explained in the task's initial description. Each node's importance within the graph is measured and each node updates its influence metadata based on the PageRank value that it has received. Moreover, through the bot detection model, it has, in line with the task description, the ability to classify authors as being humans or bots.

The model for detecting bots is a MPL neural network which is able to classify a given node based on its metadata information as a bot or human with accuracy of 83% and a weighted F1-Score of 82%. This is already well above the GA requirement to reach "at least 60%" in terms of "the capabilities of LEAs personnel in identifying sources of D&FN," given an interpretation of the word "sources" as referring to distinguishing between bots and human operators. In the event the term is meant to allude to the account spreading D&FN, the rate is even at 100%, considering that all accounts from which the posting and re-posting originates are known to LEA end-users and the spread of their messages is illustrated in the form of the above-mentioned graph. The further requirements to "increase the ability of LEAs personnel by 70% to identify who is driving a campaign"<sup>54</sup> and to achieve an "60% increase of effectiveness of the AI-based service in monitoring the D&FN actions"<sup>55</sup> (which, presumably, also alludes to the spread of such D&FN) are harder to grasp for the time being, since the LEAs' capabilities to distinguish between accounts of real persons and bots and to do spread analyses are likely to vary and remain to be inquired into in the pilots. Exact comparisons can be made then. In the absence of any such capabilities, which, presumably, applies to numerous LEAs that are at least rather unlikely to be familiar with tools that distinguish between real persons' and bots' accounts, however, the threshold of 70% has been surpassed already. The same applies to the required level of "accuracy on the assessments of the origin of D&FN", which the project's end-user survey has found out to be above 80% (see D2.1, for further information), which is fully in line with the further requirement of "[v]erification of the threats and risks identified to be related to D&FN in >80% of the case[s]."<sup>56</sup>

In addition to this first version's results plans have been made to further improve its performance by applying more advanced pre-processing techniques and also SOTA NLP methods in case they can be applied to the current dataset format.

<sup>54</sup> Ibid., the GA assigns this objective to the Behaviour Profiler but it is the Spread Analyser that analyses the sources of D&FN in the sense of distinguishing between bots and humans and doing a spread analysis.

<sup>55</sup> 'Grant Agreement: Project 101073980 – FERMI – HORIZON-CL3-2021-FCT-01,' *European Research Executive Agency*, 2021.

<sup>56</sup> Ibid., the GA originally stipulated that 95% accuracy was desirable but the FERMI consortium's end-user survey revealed that more than 80% was deemed sufficient by expert practitioners, see on this D2.1.

Finally, the current status of the aforementioned component regarding the R&I maturity is at technology readiness level (TRL) 5: technology validated in relevant environment (industrially relevant environment in the case of key enabling technologies). A very promising TRL, given that it has not yet been integrated in an operational environment, which will be included in the next plans. With a TRL of 5, the Spread Analyser is already approaching TRL 7, which is mentioned as an objective by the GA.

### **3.4 Next Steps**

#### **3.4.1 Functional advances**

The next phase of development aims to introduce further optimisations and improvements to the Spread Analyser and its components. This will be possible by experimenting with different approaches and lead to the adoption of the most optimal ones. The planned actions will focus on: (1) improving the graph building service, (2) trailing other identification methods, (3) optimising the adopted graph expansion policy, (4) enhancing the influence analyser, and (5) improving the F1-Score and accuracy of the classification model.

Future improvements to the graph builder have been considered, particularly by introducing graph updating functionality. This way the user will be able to update the graph without the need to initiate a new investigation request. The graph will be re-created from the initial user-provided social media post. This would allow for any new information on the existing nodes or new edges, altogether, to be included in the graph. As for optimising the graph expansion policy, this entails continually trialling different approaches, seeing, with their results, if the decided upon policy remains the wisest choice. Said trailing would most likely feature depth and post-depth expansion optimisations and optimising of the posts' expansion queue. Improved accuracy would be sought through different methods of pre-processing the inputted data, drawing on past attempts made in the field.<sup>57</sup>

#### **3.4.2 Integration of Further Social Media Platforms**

Integration of multiple social media platforms is feasible for the already developed current version of the Spread Analyser module. Nevertheless, this would require to consider the adaptation of the intricacies of each social media platform's API solutions. The graph-building component can create network graphs for a range of social media platforms, that is, those with clear relationships between posts. However, different platforms often use different terms for similar relationships. This necessitates tweaks in development for each platform, although the core process of building the graph remains independent of the specific platform. Also, each social media platform comes with its own set of rules and offers different types of API capabilities, which can affect how much and what type of data can be accessed. This directly impacts the functionality of the Social Media API Crawler, requiring customisation for each platform's unique APIs and their specific limitations and complexities.

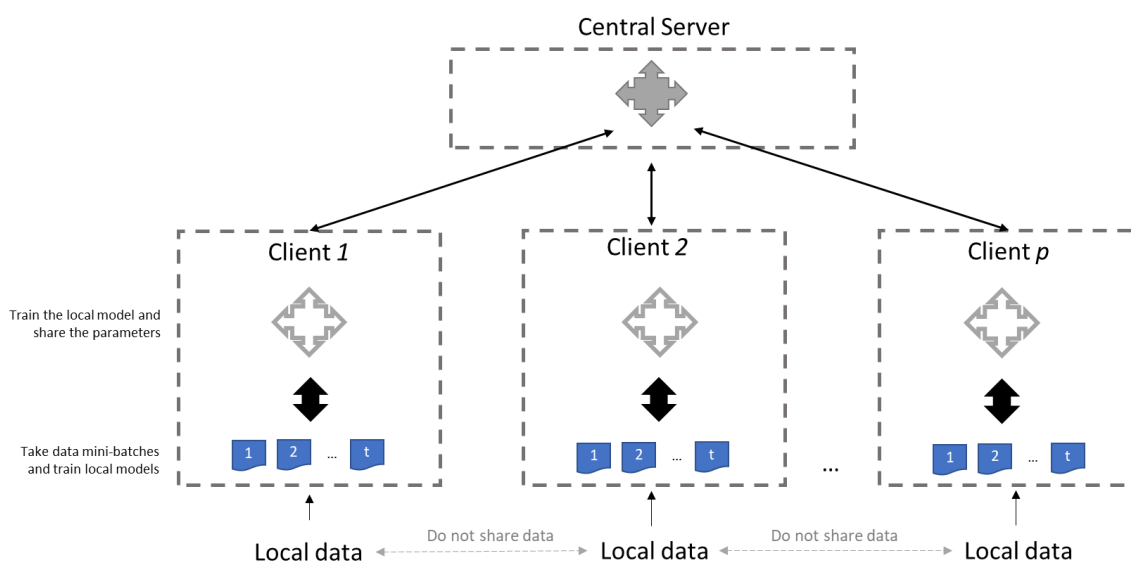
Despite these challenges with data gathering and graph building, the Influence Analyser module is quite adaptable and functions effectively across all platforms, provided it receives the necessary graphs from the graph creator component. Additionally, the bot model, initially trained using data from X, can be retrained with data from any new platforms that are integrated. This is necessary to accommodate the different types of data and formats these platforms might bring. Finally, the Orchestrator Service is designed to be independent of any particular social media platform. It handles requests for analysis, manages the flow within the Spread Analyser module, and ensures smooth communication with other dependent modules in the FERMI system, all without processing the data itself. This makes it versatile and capable of integrating with any platform. Consequently, depending on the level of access and available data, any social media platform API could be potentially integrated into the Spread Analyser module. To achieve the above, research regarding the suitability of other social media platforms' APIs is underway. Given their suitability, integration can be attempted, particularly based on end-user requests during exploitation of the platform.

---

<sup>57</sup> Ferrara, E., & Kudugunta, S., 'Deep Neural Networks for Bot Detection,' *Information Sciences*, 2018.

## 4 Task 3.4 – Swarm Learning for Holistic AI-based Services

Federated learning (FL) is a broad topic that has gained increasing attention from both industry and academia,<sup>58</sup> its purpose is to train global ML models using private data provided by several, independent agents.<sup>59</sup> **Ideal for ensuring that data-privacy is maintained**, FL is designed such that no agent should be able to make any inference about the data of any other, except for the output produced through the aggregation of all the provided data. In other words, the data of an agent, and any model trained on that data exclusively, remains private. **The overall FL protocol, sometimes named *vanilla FL*, refers to an algorithm to train a global model by aggregating local private models trained individually in the agent infrastructure.** The term *vanilla* refers to the widely adopted central server communication pattern, wherein a third-party agent (the central server) acts as a coordinator between different agents' infrastructures, aggregating the weights of all the local models. This process is repeated for several rounds in which (1) the central server broadcasts the global model parameters to all agents; (2) the agents train the local models initialised with the global model parameters, and (3) the central server aggregates the local models.



**Figure 15: The vanilla FL protocol**

Swarm learning is a specific FL protocol that builds on the aforementioned vanilla FL, differing in that it removes the need for a central server agent. Instead, swarm learning suggests shifting the coordinator role between the participating agents, through said agents electing one of themselves as coordinator. In swarm learning, a communication round is undertaken, synchronising the agents and creating agreement on a single global model. Generally achieved by securely aggregating all the local models into a single agent and then broadcasting that global model back to each agent.

Within FERMI, special emphasis was placed on the concept of agent infrastructure, a computational resource that holds private data. This is not necessarily a single computing node but a collection of computing nodes behind which private data can be accessed. For FERMI, the referenced servers/agents are those of differing European LEAs, making the protection of data privacy essential. In particular in meeting the commitments laid out in the GA, a semi-honest model was adopted.<sup>60</sup>

The semi-honest model is a security model in which all the agents involved follow the designed protocol, but the potential for negation (i.e., interest by one agent in viewing the data of another) is taken for granted. Privacy

<sup>58</sup> Chen, T., et al., 'Federated machine learning: Concept and applications', *ACM Transactions on Intelligent Systems and Technology*, 2019.

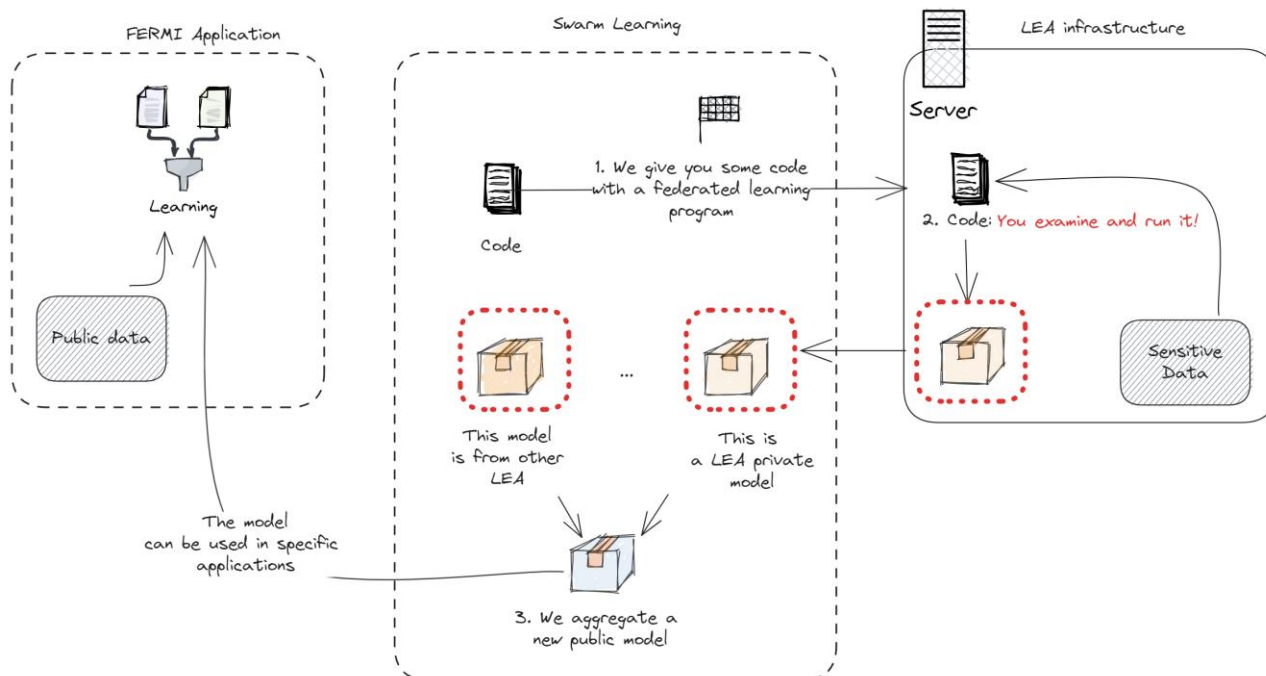
<sup>59</sup> Arcas, B.A., et al., 'Communication-efficient learning of deep networks from decentralized data'. *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, 2017.

<sup>60</sup> Goldreich, O., *Foundations of Cryptography: Volume 2, Basic Applications*, 2004.

preserving solutions were, therefore, incorporated into the constructed infrastructure. The remainder of section 4 will give a general overview of the swarm learning to be employed in FERMI and how it complies with the commitments articulated in the GA. Just as well, the current state of development and the next steps towards its completion/betterment will be discussed.

## 4.1 Practical Description

As previously mentioned, swarm learning allows for the training of global models using independent agents’ private data. For the FERMI platform, said agents are LEAs who abide by data privacy standards, thus, while justifiably needing to keep their data private, can not capture the benefits cross-jurisdiction data analysis could provide them. **Swarm learning serves as a bridge, making the benefits of said analysis achievable without any privacy being sacrificed.** In particular, the global model employed by FERMI is the Dynamics Flows Modeler, as described in section 2 and T3.1. **The Dynamics Flows Modeler, thanks to swarm learning, will be able to study past crime occurrences, in Europe, without LEAs turning over confidential data.** Figure 16 illustrates, graphically, the position of swarm learning within FERMI.



**Figure 16: Swarm learning as positioned with the greater FERMI platform**

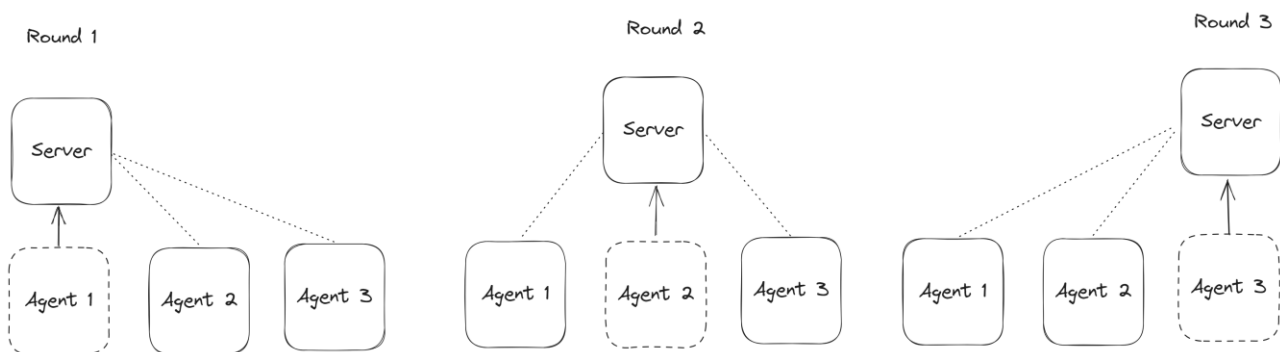
Subsection 4.1 discusses the main functions and benefits of swarm learning, as it has been structured for FERMI, and, briefly, the methods behind its operation (covered more in-depth in 4.2). The objective of T3.4, as stated in the GA, is **to develop “the software infrastructure to create a [swarm learning] framework, which will provide a scalable software architecture for training ML models near to the data sources where they are generated.”**<sup>61</sup> From this, three different components emerge: the creation of the swarm learning framework, developing its scalability, and the training of ML models close to the data source.

### 4.1.1 The Swarm Learning Framework

Swarm learning allows the agents undertaking a FL protocol to act as clients and servers, simultaneously. The server role is assigned dynamically at the beginning of each communication round, and the other agents send their local model to the elected server agent, who then forwards back to them an aggregated global model from the server. Subsequently, a new communication round begins, a new server agent is elected, and the process repeats. This process is illustrated in Figure 17, with three example agents

<sup>61</sup> ‘Grant Agreement: Project 101073980 – FERMI – HORIZON-CL3-2021-FCT-01,’ *European Research Executive Agency*, 2021.

and communication rounds. In the first round, Agent 1 becomes the server and sends the global model to both Agent 2 and Agent 3. Once those agents finish training that model, they send the resulting local models back to Agent 1, completing the first FL round. After this, Agent 1 will send the resulting global model to Agent 2 who will assume the role of server in the next round. The same process will take place with Agent 3 acting as central server, and the process will continue successively.



**Figure 17: An example of swarm learning with three agents and three communication rounds**

Swarm learning has several functional and non-functional requirements, derived from the structure specified above: (FR1) coordinate server selection, (FR2) aggregation of local models in a single agent, (FR3) submission of local training request that, given a parameter vector, returns an updated version of model parameters after local optimisation, and (FR4) the provision of a mechanism to specify the identity of different agents participating in the FL protocol. In addition, the sole non-functional requirement (NFR1) wherein the execution time needs to be scaled linearly with the number of federated agents to ensure the completion of given communication round.

#### 4.1.1.1 Coordinate Server Election (FR1)

In swarm learning, agents need to synchronise at the beginning of each round to aggregate the local models. This process involves selecting an agent to act as central server. From a security standpoint and assuming a semi-honest model, each client should have the same chance of being the server. This way, the central server aggregation will be conducted following the round-robin policy between the different clients.

#### 4.1.1.2 Model Aggregation (FR2)

At any given time, an agent can act as a server depending on the result of the FL round server election. When an agent acts as a server, it is responsible for receiving the local models from the other agents, aggregating the received local models into a single global model and broadcasting such global model to all agents. Furthermore, each agent can act as a client responsible for receiving the global model from the server, train that global model for a given number of training steps and further send the updated model back to the server. The resulting derived low-level functional requirements are enumerated as follows.

##### 4.1.1.2.1 Reception of Local Models

When an agent is acting as a server, it is responsible for aggregating the local models from the agents. There are two faulty scenarios the server considers: (1) a client never sends weights and (2) the client sends corrupted weights. In the first scenario, the server must wait for all clients, up to a maximum time. After that time, the server assumes that the agent is in error state and will continue the aggregation discarding that agent model. In the second scenario. The server should verify if the weights sent by an agent are valid or not (for example, a wrong number of parameters, or an unexpected request from an agent). If the model parameters' validation fails, then the server assumes that the agent is in error state and continue the aggregation.

#### 4.1.1.2.2 Aggregation of Local Models

When the server receives a local model's parameters, it proceeds to aggregate the parameters with the other agent's local model parameters. This is achieved by conducting the unweighted model aggregation mechanism, such as the mean of all model parameters.

#### 4.1.1.2.3 Broadcasting of Global Model

After the aggregation of all model parameters is achieved, the server sends the global model to all agents in a round robin fashion sequentially. If an error occurs while sending the global model parameter to an agent, the server will perform several retries up to a maximum number of retries. After a failed attempt of sending the global model parameter, the agent will proceed with the next agents before retrying with the failed agent. The server must implement an additional exponential *back off* mechanism, so a prudential amount of time passes by before two consecutive retries. The amount of time before the next attempt is given by  $t = b^c$  where  $b$  is a configurable parameter and  $c$  is the number of previous failed attempts. If the maximum number of attempts for one or more agents is reached, the server will assume they are faulty and will finish the aggregation round by sending a notification to all healthy agents, signalling that the FL round is finished and sending a list of agents that participated successfully in the round.

#### 4.1.1.2.4 Reception of Local Model from Server

When an agent is acting as a client, it waits for the global model parameters from the central server. Each agent waits until a maximum amount of time before declaring that the server has failed. When an agent's timeout is reached, the agent will send a notification signal to all the other agents to start a new fresh FL round shifting the server role to the next available agent, triggering a new server election round and discarding the faulty server from the FL protocol. The client should verify that the weights sent by the server are valid (for example, the incorrect number of parameters, or an unexpected request from an agent). If the model's parameters' validation fails, then the client assumes that the server is in an error state and triggers a new FL round.

#### 4.1.1.2.5 Sending Updated Local Model to Server

After completing the local training, a client sends the resulting weights to the central server. Before sending the weights to the central server, an agent will wait for a random amount of time (between a minimum and a maximum configurable value) to prevent throttling the server. An agent will wait until the server notifies that the FL round broadcast is completed before sending the updated weights to the server.

#### 4.1.1.3 Local Training (FR3)

When a global model is received from the central server, each agent acting as a FL client will take the model parameters and optimise them for a given number of epochs. The weights will correspond to a neural network model specified in a Keras v3 serialisation format. That model should be provided by the user of the framework beforehand.

#### 4.1.1.4 Agent Identification (FR4)

In FERMI's swarm learning infrastructure, different participating agents will be identified through a public key method. Using a public method makes it such that the user is responsible for providing public key certificates for the agents allowed to join the protocol.

#### 4.1.1.5 Swarm Learning as Scalable Software Architecture (NFR1)

Scalability is the capability of a system to handle a growing amount of work.<sup>62</sup> In our context, scalability means achieving a swarm learning implementation that is capable of handling an increasing number of participating agents. To this end, we define NFR1 as the execution time needed to complete a single FL round and believe that this time should scale linearly with the number of agents. To fulfil this, a background tool called Fleviden was leveraged, making software architecture scalable. Fleviden is a Python library created by the Research and Development Department of ATOS. It is used to develop FL algorithms, in general, that was extended to support the unique swarm learning requirements of FERMI.

#### 4.1.2 Training ML near to Data Sources

The aim of FL, including swarm learning, is to implement a specific ML model in a defused manner. In the case of FERMI **the implementation algorithm aims at supplementing the Dynamic Flows Modeler by forecasting the number of different crime types in the three countries, where pilots will be held, with LEA consortium members' data, D&FN aside, and the creation of a dataset of past crime occurrences.** The former exhibits the function of the swarm learning technology, while the latter is its current application to the FERMI platform. To align with the Dynamic Flows Modeler, and the greater FERMI platform as a whole, the swarm learnings output has the following characteristics. Estimates will be made for Belgium, Finland, and Germany, the countries to be featured in the pilots. The level of analysis, for both past crimes and estimates, will be the NUTS2 regions in said countries. Temporarily, forecasts will be made weekly, and 11 crime types will be predicted (see Figure 6), as is the case of the Dynamic Flows Modeler.

Thus far, crime occurrences have been collected from LEAs affiliated with FERMI: the Bavarian University of Public Services' Police Academy (BPA), for the years 2012-2022, the Finnish Ministry of the Interior (FMI), 2019-2022, and the Belgian Federal Police (BFP), 2010-2022. ML models have been trained using said data, **aligning with the GA commitment to use data provided by LEAs, particularly, "records of criminal events."**<sup>63</sup> It must be noted that these models can be retrained in the future, and even at regular intervals, whenever data that better suits the end-user becomes available, that is, whenever the LEAs are able to provide more accurate crime data specific to their regions or countries. The ability to update and fine-tune the models with new data will continuously enhance the accuracy and relevance of the predictions, ensuring that the analytical tools remain aligned with the evolving dynamics of criminal activity. This proactive approach guarantees that decisions and strategies based on these models are always informed and pertinent, thereby optimising efforts in crime prevention and response.

The provided datasets differed in terms of crime types, at the time of this deliverable's authoring, the FMI crime classifications have been adopted in the ML model training, while the data passed to the Dynamic Flows Modeler is fitted to the American universal crime reporting system's categories. FMI's data was complemented by socio-economic controls, as done for the Dynamic Flows Modeler. Population structure, education level, disposable income levels (individual and familial), household composition, building structure (average floor area, number of residential buildings, amount of other buildings, etc.), industry's presence, and main activity (employed, unemployed, students, pensioners, etc.) were included in the forecasting models.

A limitation arose, however, regarding the data collected for Germany. Due to the federalised police system in Germany, only the crime data for Bavaria, where BPA is located, could be collected. **Efforts are currently underway to expand the dataset's coverage** in Germany. Moreover, Belgium and Germany have not had socio-economic controls collected, which are not processed/maintained by the police, although said data's collection is progressing.

<sup>62</sup> Bondi, A.B., 'Characteristics of scalability and their impact on performance', *Proceedings of the 2nd international workshop on Software and performance*, 2000, pp. 195 – 203.

<sup>63</sup> 'Grant Agreement: Project 101073980 – FERMI – HORIZON-CL3-2021-FCT-01,' *European Research Executive Agency*, 2021. More specifically, the GA stipulates that "[t]he use of federated learning will ensure privacy and protection of personal data as well as Police Authorities autonomous control of the data. Through the adoption of a federated learning strategy, the data management will be articulated in order to guarantee end-users' full control of data provided by them (e.g., records of criminal events) while guaranteeing the full protection of privacy and security of the data."

As a work of feature engineering, all the features extracted from the “date” column (week, month, week of the year, etc.), were transformed into cyclical variables to make it easier for the ML model to map the features of the predicted labels (number of crimes). This transformation usually allows us to obtain better results in the forecasts, as some periodic patterns could not be easily found by the ML algorithm. The variables were encoded using sine and cosine methods. Once the data cleaning process was finished, we have applied a predictive algorithm to obtain the number of crimes committed for the different types of crimes, the different regions of Finland and a specific number of weeks. The results and metrics are shown in the following section.

## 4.2 Technical Description

In this subsection, we will describe the methodology and technicalities of the practical description, explained above, at greater length. Firstly, this applies to **the Fleviden tool** the swarm learning is based on, then a description of how it has been adapted to launch the swarm learning-specific solution will be provided. The design solutions main features, and the preliminary results will also be covered.

In the GA, there is the specific requirement that the technology be extended and improved to achieve the successful implementation of the swarm learning paradigm. **The GA commits the FERMI consortium to working with “a completely decentralised approach which will guarantee compliance with existing regulations for data protection and minimise the attack surface,”<sup>64</sup> while facilitating “the dynamic and agile collaboration between multiple LEAs throughout the European geography since the role of a central entity will be not needed.”<sup>65</sup>** Just as well, the GA states that the FERMI consortium enables the “onboarding of the nodes or agents that will participate in the framework and for sharing the learnings in a safe and secure manner.”<sup>66</sup>

### 4.2.1 The Fleviden Tool

Fleviden is a fully extensible FL framework originally developed internally by the Research and Development Department of ATOS. It was included as part of the background technologies described in the FERMI’s Consortium Agreement and has been given, for FERMI, the specific objective of extending and improving the technology to implement the desired swarm learning paradigm. **The main architectural pattern of Fleviden is pipes and filters, and its core component is the pod.** A pod implementation ignores the distributed aspect of the FL logic by only focussing on small pieces of functionality,<sup>67</sup> for example, the internal aggregation of the federated server or the masking transformation performed in some of the secure-sum protocols.

A pod defines wires to interact with the exterior,<sup>68</sup> another term for the listener/observer design pattern. **The pod’s wires are, essentially, a contract interface with other pods.** There are two types of wires, input and output interfaces. Both are registered in the same way, that is, by using the Pod.register method. The main difference between an input and an output wire is that the pod itself provides a default handler in the case of input interfaces, a function, to process the message sent through said wire. This transforms the message before forwarding it through an output wire, if needed. Conversely, an output wire does not have a default handler, but it is expected that other pods register to the output wire, forming a pipeline.

**Everything in Fleviden can be described in terms of pods.** Custom functionalities are provided by just extending pods; for example, in a client-server federated communication protocol, we may have a client pod and a server pod. Within FERMI, the Fleviden tool is being extended and improved to implement the proposed swarm learning component. In doing so, it meets the GA’s above mentioned commitment to provide “a completely decentralised approach which will guarantee compliance with existing regulations for data protection and minimise the attack surface,”<sup>69</sup> facilitate the “the dynamic and agile collaboration between

<sup>64</sup> Ibid.

<sup>65</sup> Ibid.

<sup>66</sup> Ibid.

<sup>67</sup> Mira, J., et al. ‘D3.1 Federated Learning implementation’, *ALCHIMIA Horizon Europe Project*, 2023.

<sup>68</sup> Ibid.

<sup>69</sup> ‘Grant Agreement: Project 101073980 – FERMI – HORIZON-CL3-2021-FCT-01,’ *European Research Executive Agency*, 2021.



---

multiple LEAs”<sup>70</sup> and enable the “onboarding of the nodes or agents that will participate in the framework and... sharing the learnings in a safe and secure manner.”<sup>71</sup> In the following subsections we describe the proposed solution, drawing a clear line between the already existing functionalities, provided by Fleviden, and the new, extended ones.

#### 4.2.2 A Swarm Learning Solution in Fleviden

As illustrated in Figure 18, a class diagram, a solution to FERMI’s swarm learning is characterised by a part of the larger Fleviden library, in particular the already existing Pod, Keras, Server, and Mask classes. Said classes were placed in the core, trainers, cen, and privacy packages, used to define new pods, train models, and provide additional privacy preserving mechanisms, respectively. In the development of the solution to be used **in FERMI, new pods were created to support the swarm learning’s functional requirements, as described in 4.1.** Special attention should be paid to the class pod that is placed in the core of the Fleviden tool. Importantly, the new classes are Asymmetric, Agent, and Rotator, derived from this base class. The Agent class is composed by the original Server pod and the Rotator pod extending their functionality and reusing already existing software.

---

<sup>70</sup> Ibid.

<sup>71</sup> Ibid.

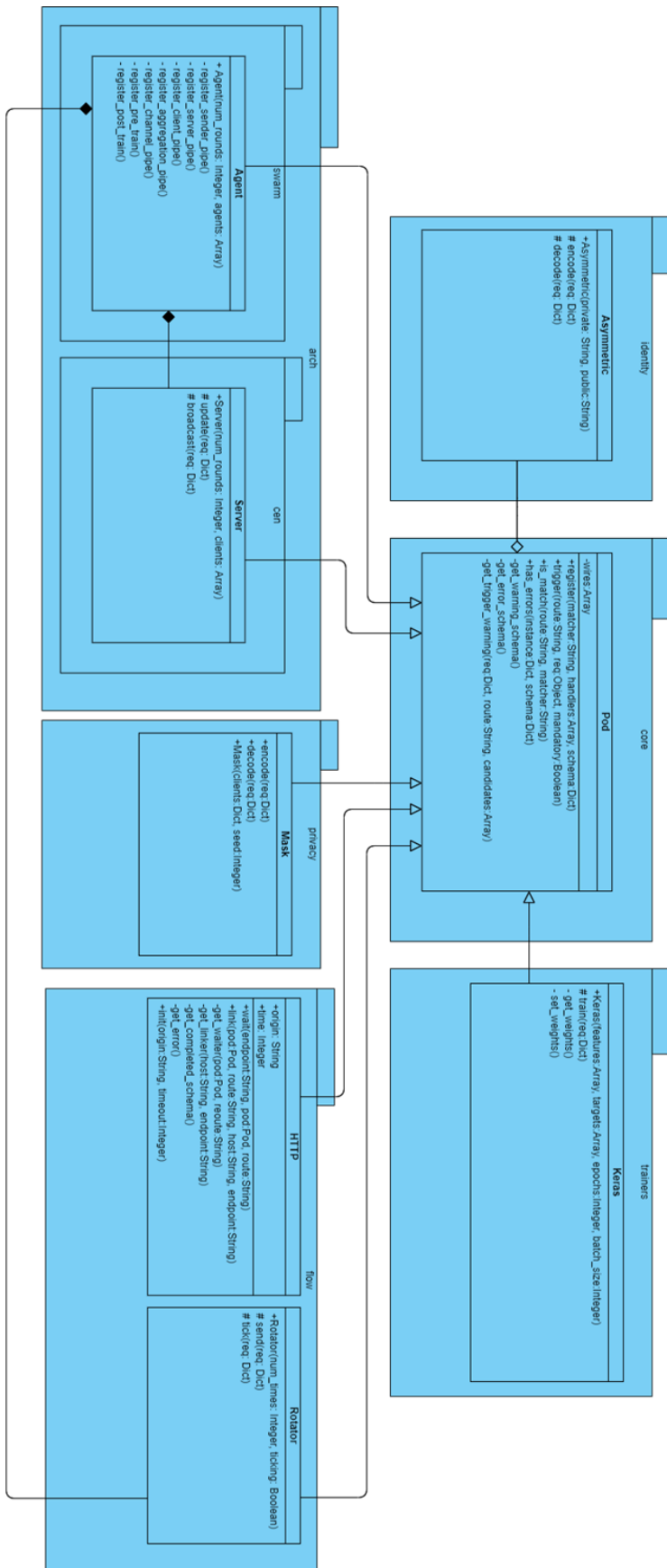


Figure 18: Class diagram of the Eviden tool. The main relations between the new pods introduced and the existing ones

There is a direct relation between the provided pods and the functional requirements specified for the development of the swarm learning components. FR1 and FR2 are provided by the Agent pod that in turn delegates most responsibilities to the Server and Rotator class. FR3 is provided by the Keras pod and FR4 is provided by the Asymmetric pod. For completeness, in what follows we provide a careful description of the different pods, their methods and responsibilities.

**Table 2. Description and methods of the different Pods from Fleviden’s framework**

Pod Name	Description	Methods
Pod	A single and self-contained piece of functionality encapsulated behind a well-defined public interface defined by input and output wires. It can be connected to another Pod such as a pipeline, which is created by filtering and enriching the messages transferred from Pod to Pod.	register(self, matcher, *handlers, schema=None): Registers a new wire, it can be an input or output interface/wire as in practice there is no difference between them. The difference lies in whether the pod provides a default handler or handlers for the interface.trigger(self, route, req, mandatory=True)
		trigger(self, route, req, mandatory=True): Triggers a message to be processed by the pod or to be forwarded via an output interface.
		is_match(self, route, matcher): Checks that the given route matches the provided matcher. This check is currently implemented as a simple equality but in the future, there may be a more complex matching functionality, e.g., variable parameters in the route.
		has_errors(self, instance, schema): Validates that the given instance object matches the provided schema. The schema should follow the “json schema” validation library definitions. <sup>72</sup>
Keras	Responsible for training a neural network model for some number of epochs using a provided training data.	train(self, req): Handler for the /train interface responsible of triggering a new training round on the specified Keras model.
		get_weights(self): Given a keras model, it obtains the model parameters in a linearised one-dimensional list.
		set_weights(self, weights) Given a keras model and a linearised list of floats, it sets the model parameters from the given list of floats.
HTTP	This class allows to connect a pod output wire to another pod via HTTP protocol by specifying a host endpoint server.	wait(self, wire): Registers a new HTTP REST endpoint that is linkable to an output wire and other pods.
		bridge(self, route, host, endpoint): Connects an input route / wire to another REST endpoint.
Mask	This class implements a secure-sum protocol. The protocol assumes a client-server FL architecture such as each client is assumed to be honest, and the server is assumed to be adversarial.	encode(self, wire): Handler for the /encode input interface that deals with encoding a parameter vector.
		decode(self, req): Handler for the /decode input interface that deals with receiving the private aggregation from the server.

<sup>72</sup> See <https://json-schema.org/>.

Server	A server-side implementation for federated learning central server aggregation.	update (self, req): Receives weights from client. Check that the origin of the weights is registered in the list of clients.
		broadcast (self): It sends the currently stored weight aggregation to the registered clients for further aggregation
Asymmetric	A pod for cyphering and deciphering messages using an asymmetric public key infrastructure.	encode (self, req). Cypher the incoming message using a private certificate and send the encoded message in the corresponding interface.
		decode (self, req). Check if the received message is signed by a given public certificate and send the plain message to the corresponding interface.
Rotator	This class allows to specify several wires that get created and triggered in different stages in a round-robin fashion.	send (self, req): Send the message through all output interfaces.
		tick (self, req): Send the message through the next output interface.
Agent	Implement a swarm learning agent that can coordinate with other agents to train a global model in a federated learning fashion. Swarm learning is a protocol in which several agents that have private data train a global model with the role of the central server rotating.	register_sender_pipe (self). This method registers the wires responsible for broadcasting the global to model all agents when this agent acts as a server.
		register_server_pipe (self, req). This method registers the wires responsible for receiving the model weights for all agents, aggregate them and forwards the result to the next selected agent who will act as server in the next round.
		register_client_pipe(self). This method registers the wires responsible for receiving the global model from the server, starts training it and sends it back to the server.

### 4.2.3 Design Solution’s Main Features

The current FL / swarm learning solution safeguards the confidentiality and security of personal information, while also granting LEAs independent command over the data. By embracing the proposed solution, data administration will be structured to empower LEAs with complete authority over the information they provide, such as crime incident records. This is possible due to the use of a decentralised approach in which several agents that have private data and train a global model rotate the role of the central server. This strategy ensures not only the protection of privacy but also the security of the data. As stated in the GA, “the use of [FL] will ensure privacy and protection of personal data as well as [LEAs] autonomous control of the data. Through the adoption of a [FL] strategy, the data management will... guarantee end-users’ full control of data provided by them (e.g., records of criminal events) while guaranteeing the full protection of privacy and security of the data.”<sup>73</sup> The framework supports the most widely used deep learning frameworks such as Keras with a TensorFlow backend, again adhering to the GA, which describes T3.4’s framework as one which “will support [the] most widely used deep learning frameworks like TensorFlow, PyTorch or Caffe”<sup>74</sup>

The agents involved in the protocol assume a given index as their identities vary from *zero* to the *number of agents*, that is, one. This index is used to assume the role of the abovementioned central server in charge of aggregating the local models from the other agents. This initial solution does not implement any complex protocol to determine which agent acts as a server in each round. Instead, as explained above, the agent that acts as a server is selected in a round robin fashion, based on their indices. Regarding the technical requirement stated in section 1.2.2.1 of the GA, “a permissioned blockchain network will be used for the

<sup>73</sup> ‘Grant Agreement: Project 101073980 – FERMI – HORIZON-CL3-2021-FCT-01,’ *European Research Executive Agency*, 2021.

<sup>74</sup> Ibid.

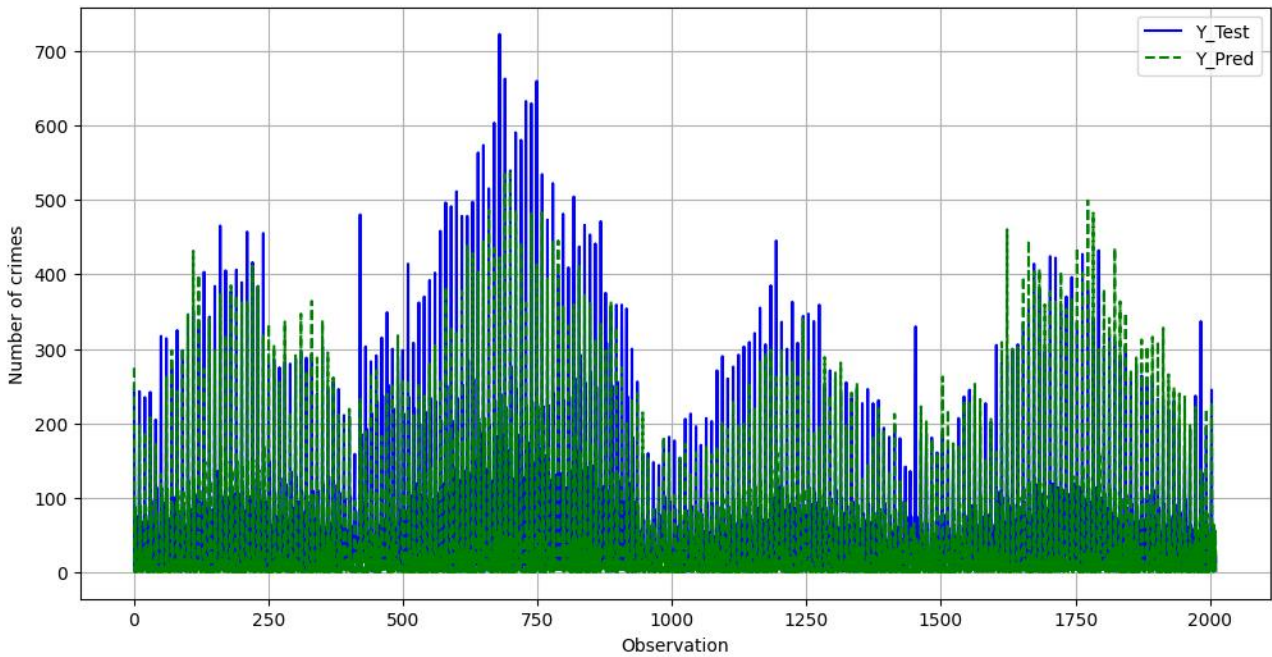
onboarding of the nodes or agents that will participate in the framework and for sharing the learnings in a safe and secure manner,” blockchain implementation has been studied and it could imply deploying additional software and infrastructure, while applying a cryptographic mechanism based on public and private keys is enough to authenticate the different agents / LEAs which will participate in the federated protocol, making it such that the functionality will be covered the same.

Another technical requirement made in said section of the GA, promising the use and integration of SOTA NLP libraries (e.g., Hugging Face), unfortunately, does not fit the swarm learning component as no texts are being analysed for its infrastructure. Instead, the swarm learning is oriented towards the processing of crime incident data from participating LEAs. By its nature, social media and other online content are not private and, therefore, do not require the privacy-protecting infrastructure of swarm learning for processing. That being said, the Sentiment Analysis module has been developed with the help of Hugging Face, amongst other things.

#### 4.2.4 Preliminary Results

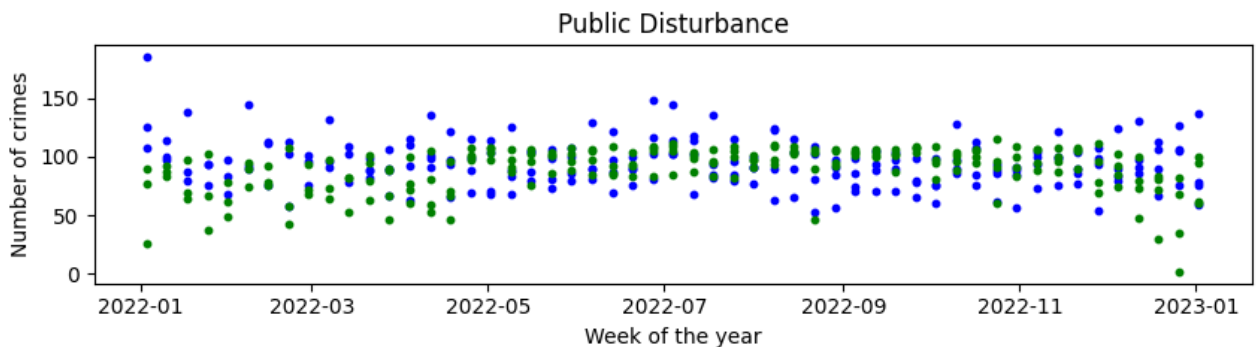
The results presented here are those of the ML forecasts produced using the swarm learning infrastructure. Thus, they underscore the successful development of swarm learning, between agents, and the ability to train ML models close to the data source. **A long short-term memory (LSTM) ML model was chosen**, as a recurrent neural network it **allowed for the appreciation of long-term dependencies in the provided sequential data** (similar to how the neural networks 1-D CNN and transformers perform in the Dynamic Flows Modeler). FMI data was employed, containing **10 crime types**, after being split into a training and test set. The deviation between the crime types included here and in the Dynamic Flows Modeler is due to the difference between how the FMI and Federal Bureau of Investigation grouped criminal acts into crime categories. Where **the years 2019 – 2021** were demarcated as for training, while 2022 served as the test set. the algorithm is able to predict about 2000 observations of time (10 type of crimes × 4 regions in Finland × 52 weeks/year). It must be noted we have only 4 regions in this case because we do not have actual data from one of the Finnish NUTS-2 regions (FI20 - Åland).

As with the Dynamic Flows Modeler, MAE was chosen as the metric by which performance is to be measured, with root mean square error (RMSE) also providing insights with respect to accuracy. Currently, performance is at a MAE of 29.81 crimes and a RMSE of 48.12 crimes. In Figure 19, all the data available in the different regions, crimes and weeks from Finland are depicted. The blue line corresponds to the real number of crimes and the green line corresponds to the estimated number of crimes inferred by the LSTM algorithm. As it can be observed, the tendency is well detected by the algorithm.

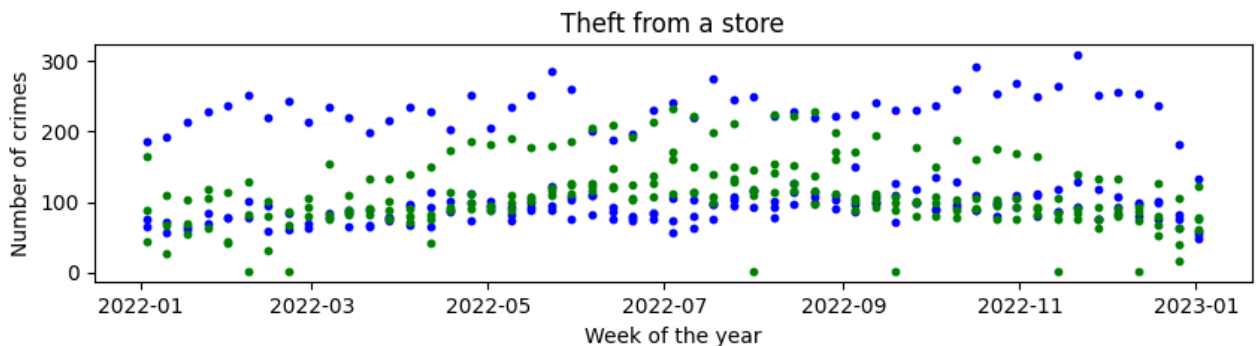


**Figure 19: LSTM crime forecasts, conducted with swarm learning**

The estimates obtained for different type of crimes are shown in Figures 20, 21, and 22. The first refers to public disturbance, the second to disorderly conduct, and the last one to shoplifting (theft of belongings from a store). For each week, 4 points are depicted as there are 4 NUTS-2 regions.



**Figure 20: LSTM forecasts, conducted with swarm learning, for public disturbance**



**Figure 21: LSTM forecasts, conducted with swarm learning, for disorderly conduct**

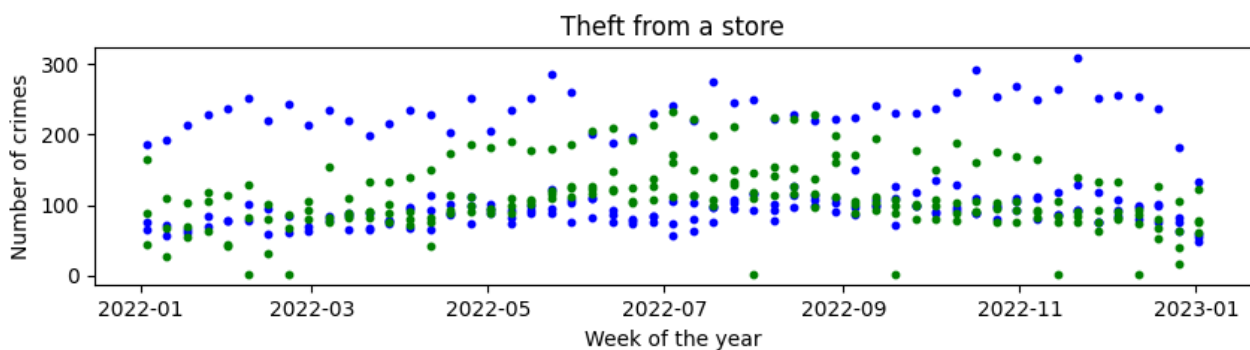


Figure 22: LSTM forecast, conducted with swarm learning, for theft from a store

### 4.3 Current Advancement and Demo

In 4.2, the offerings and objectives of the swarm learning framework articulated in the GA will be reviewed, with respect to their alignment with what has been accomplished by FERMI thus far, and where additional effort is to be made to ensure compliance in the future. In particular, the GA commits the swarm learning protocol to being able to “facilitate training ML models for predicting offline and online crime caused by D&FN, tailored to the specific needs of police authorities;”<sup>75</sup> use “a completely decentralised approach [which] will guarantee compliance with existing regulations for data protection and minimise the attack surface;” “facilitate the dynamic and agile collaboration between multiple LEAs throughout the European geography since the role of a central entity will be not needed,” and increase “SOTA learning speed by at least 50%.”<sup>76</sup>

Regarding the facilitation of ML model training, to make estimates of offline crime given online D&FN, **the swarm learning framework is proven, as exhibited above, to be able to make crime forecasts and, as articulated in section 2, provide integral past crime incident data to the Dynamic Flows Modeler,** which allows for the estimates of crimes given an online D&FN event. As for using a completely decentralised approach, the developed swarm learning framework, in this task, ensures a quicker collaboration between LEAs, as mostly the main bottleneck is the sharing of sensitive data between them. With the current demo based on Docker containers, the feasibility of implementing a FL approach is demonstrated, which avoids the sharing of data and minimises the attack surface as there is not a fixed server where all the data and / or insights from the data are stored. **Agencies interested in training a common and up-to-date ML model in collaboration with LEAs from other regions can easily adapt the framework with the comprehensive documentation that will be provided in the future.** By utilising the latest crime data and continuously updating the model, these agencies can ensure that their predictive tools remain accurate and relevant.

**Importantly, said quicker collaboration, when compared to other SOTA alternatives, such as vanilla FL, is more fault tolerant as there is not a single point of failure (i.e., one server).** This makes the FERMI developed swarm learning framework quicker than others, as when an error does occur, such as the FL server no longer being viable, alternative FL frameworks are incapable of handling it.

The GA also proposes that T3.4, using a complete decentralised approach to training ML models, will increase the predictive capabilities of offline and online crimes introduced by D&FN by more than 40%. The performance improvement in terms of predictive capabilities derives from the combined training using data from several LEAs in contrast to building models using just local data of each LEA separately. In addition, the fact that we are including public data as forecasting variables also improves the capabilities of our solution. **This, combined with the Dynamic Flows Modeler, represents a great advance in the forecasting capabilities of LEAs, with respect to online D&FN events impact on offline crime.**

<sup>75</sup> Ibid.

<sup>76</sup> Ibid.

---

## 4.4 Next Steps

Though the preliminary version of the T3.4, the swarm learning infrastructure, has been well developed in the early months of FERMI, and, in a sense, **there is a first, functional framework able to manage and deploy federated training in different agents**, there are certain objectives still pending completion. In **the short term, alignment between the FMI, BPA, and BFP datasets has to be completed**. When applying a FL approach, the data of the different clients / hosts must present the same features (columns) and targets (labels). It is still pending, as pointed out in other sections, to map the crime types present in the different datasets provided by the LEAs. **Moreover, the socio-economic controls for Germany and Belgium must be collected and adapted to the one already collected from Finland, which has been time-consuming due to bureaucratic constraints with the consortium's LEA not being in the driver's seat but depending on the input of numerous other government agencies**. This feature will allow us to predict the number of crimes in all the NUTS-2 areas present in the different pilots. Once all the data from the different LEAs is ready to be processed, the current regression algorithms will have to be retrained. Some effort will be needed as it is possible that some parameters and hyperparameters of the algorithms must be adapted to extract the best possible model from the available data. **Another objective that will be advanced in the coming months is that of better aligning T3.4 with T3.1, as the output of the swarm learning framework has to be adapted to the Dynamic Flows Modeler**. Crime types forecasted (and, therefore, which past crime types to provide), time periods to be covered, as well as other integration decisions need to be made. The comparison with other SOTA algorithms is pending. When the component is fully completed, we will compare it with other vanilla FL algorithms to check to what extent we have reached the GA requirement of “swarm learning mechanisms increasing SOTA learning speed by at least 50%.” **The Fleviden technology is still currently under development by ATOS. Thus, it is constantly evolving beyond its current, early stage. If advancements in the Fleviden tool represent potential improvements for FERMI's swarm learning technology, the available features will be integrated into it.**



---

## 5 Task 3.6 – The Sentiment Analysis Module

### 5.1 Practical Description

The Sentiment Analysis module is designed to assess the emotional disposition of the social media authors with respect to crucial antecedents, thereby facilitating the identification of potential linkages between the spread of D&FN in the online realm and the broader risks of offline escalations and criminal behaviours. Its technological foundation is comprised of SOTA NLP and ML. By scrutinising sentiment patterns embedded in social media content, the module can unveil linguistic nuances employed by individuals involved in the dissemination of D&FN and, consequently, constitutes a significant stride in the endeavour to assess the likelihood of D&FN-enabled offline actions, especially with respect to criminal activities.

The module harnesses the power of the cutting-edge BERT language model, which is in line with the GA's requirement to "exploit the BERT model [...] with a wide variety of NLP tasks,"<sup>77</sup> to delve into the vast realm of social media data (e.g., X posts' data graphs with highly influential nodes spreading disinformation). The BERT model leverages context from both past and future words to make an estimate for a certain task. As further stated in the GA,<sup>78</sup> experimenting with a bidirectional LSTM as a feature extractor was carried out. These two approaches will allow for the module to capture long term dependencies of the posts, to understand sentiment. The module unveils the sentiments embedded within these textual treasures. Due to the complex nature of social media data, related to D&FN, it is necessary that the module takes advantage of different aspects to precisely and accurately predict a given D&FN's content's sentiment polarity. As will be explained, in a two-phase process, namely the training phase and the inference phase, the module is able to unveil the sentiments embedded within X posts interacting with the end-user provided D&FN.

### 5.2 Technical Description

The primary objective of the sentiment analysis task is to analyse text to predict its polarity. Within the context of FERMI, the sentiment analysis module will specifically examine social media posts. The implementation of the module follows a dual-phase operational approach, comprising the training phase and the inference phase. During the training phase, the module's model undergoes training on annotated data to acquire an understanding of sentiment patterns. In the inference phase, the trained model is applied to fresh textual data to unearth embedded sentiments within the text. This functionality facilitates automated sentiment analysis, rendering it a valuable module for discerning public sentiment trends in the realm of social media.

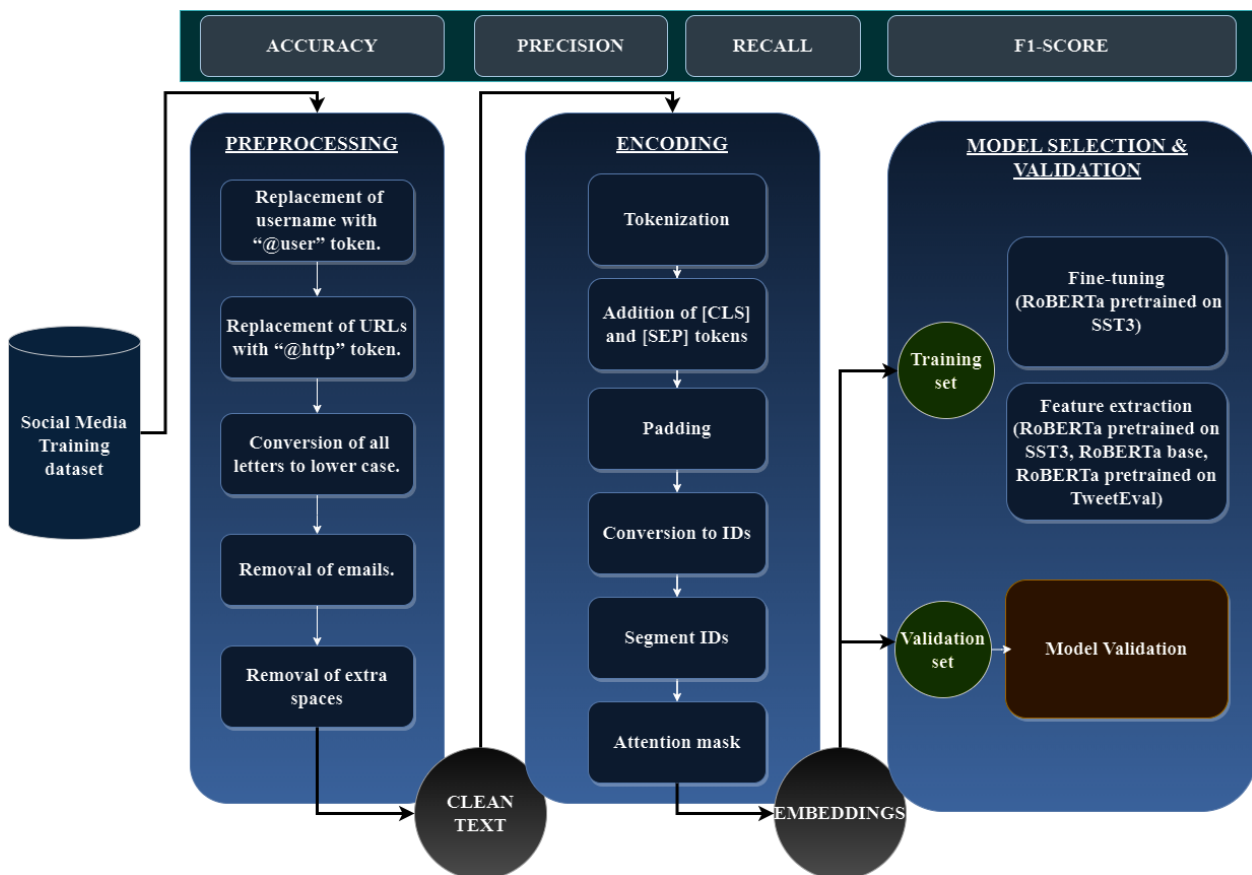
#### 5.2.1 The Training Phase

Given the intricate nature of social media data associated with D&FN, it is evident that the proposed model necessitates multifaceted considerations to achieve precise and accurate classifications of the sentiment polarity of given content. The subsequent section expounds upon the training phase steps of the fine-tuned model, providing a comprehensive overview of the process and its constituent elements.

---

<sup>77</sup> 'Grant Agreement: Project 101073980 – FERMI – HORIZON-CL3-2021-FCT-01,' *European Research Executive Agency*, 2021.

<sup>78</sup> Ibid.



**Figure 23: Step-by-step overview of the Sentiment Analysis module’s training phase**

### 5.2.1.1 Training Dataset

The selection of datasets is a critical determinant in the training of a machine learning model, fundamentally impacting its performance and accuracy. High-quality and relevant datasets, i.e. in our case datasets resulted by official research which include annotated posts acquired from the respective social media platform, enable the model to learn from representative examples, thereby capturing the intricacies and diversity inherent in real-world data. When the training data closely aligns with the target application, the model's ability to generalise and make accurate predictions is significantly enhanced. Consequently, meticulous consideration and selection of appropriate datasets are essential for the success of any machine learning endeavour. Hence, the initial phase of developing the Sentiment Analysis module entailed a considerable effort to identify and select datasets most relevant to our use case.

**Initially, the training revolved around employing the Stanford Sentiment Treebank (SST),** as required by the GA,<sup>79</sup> a corpus characterised by fully labelled parse trees encompassing 11,855 individual sentences sourced from movie reviews. This corpus includes a total of 215,154 unique phrases, each annotated by three human judges. Different versions of the SST dataset are available, such as SST-5 (SST fine-grained), in which each phrase is categorised into negative, somewhat negative, neutral, somewhat positive, or positive sentiments. Additionally, there is SST-2 (SST binary), which is commonly used in binary classification experiments focusing on full sentences. **However, SST appeared to be sub-optimal for the specific D&FN context, as it was comprised solely of one type of text, movie reviews.**

Unlike movie reviews, **social media posts contain their own distinct language characteristics. In general, each social media platform features unique linguistic traits,** thus a module trained on movie reviews would exceed the linguistic scope of FERMI. In one of the two testing strategies, a model pre-trained using SST, that is, having acquired sentiment rules from movie-related content, was used; subsequently, the model needed to be fine-tuned using a dataset of social media posts to ensure it aligned more closely with the specific linguistic nuances found in the target data. The wide range of validated and high-quality datasets

<sup>79</sup> Ibid.

including X posts, led us to utilise these datasets for our use case. Hence, to facilitate model training and benchmarking, the TweetEval dataset<sup>80</sup> was chosen as the primary corpus. It is crucial to emphasise that fully adapting our model to other social media platforms requires validated, annotated, and high-quality datasets containing posts from those specific platforms. The remainder of the development process would mostly remain the same in such cases.

The TweetEval dataset comprises texts extracted from X posts and categorised into three distinct sentiment classes: 0 for negative sentiment, 1 for neutral sentiment, and 2 for positive sentiment. The dataset is thoughtfully partitioned into three subsets, including a training set with 45,615 data points, a validation set consisting of 2,000 data points, and a testing set encompassing 12,284 data points. Importantly, **the dataset exhibits a high degree of completeness with no missing values and an exceedingly low rate of duplicate data points, at just 0.06%**. It is worth mentioning that there is a noticeable class imbalance within the dataset, where the neutral class is overrepresented while the negative class is underrepresented.<sup>81</sup>

In the context of the ML experiments undertaken, the challenge posed by class imbalance within the training dataset has been duly acknowledged and targeted for mitigation. The overarching goal of these experiments is to assess the potential improvements in the model's performance achieved through the deployment of different strategies. Specifically, three discrete approaches have been meticulously examined and are listed as follows: (1) oversampling the underrepresented classes, (2) weighted loss during training, and (3) supplementing the dataset with additional data. The first involves generating duplicate instances within the underrepresented classes, while the second, rather, employs weighted loss functions to assign increased significance to the underrepresented class. The third would, effectively, eradicate the imbalance issue by adding observations of the underrepresented class from another dataset, specifically the t4sa dataset.<sup>82</sup> The following subsection, which delves into data pre-processing, encoding, and model selection and evaluation steps, offers an in-depth examination of the results obtained through the application of these strategies.

### 5.2.1.2 Pre-processing Data

The data preparation plays a pivotal role in refining the input data for the ML endeavours and involves a meticulous cleansing process, possibly removing extraneous elements such as stop words, punctuation, and even emojis, thereby homing in on the essential textual content. Nonetheless, it is essential to emphasise that thorough data cleansing does not universally result in enhanced model performance. It is noteworthy that our comprehensive literature review failed to uncover cases of extensive text purification preceding the utilisation of BERT-based models. **The pre-process pipeline encompassed several steps, namely (1) replacement of username with “@user” token to eliminate any referenced users in the tweet and fully anonymise the text (in full compliance with data protection standards, see D7.1 and D7.2 for further information on this); (2) replacement of URLs with “@http” token to remove all URLs; (3) conversion of all letters to lower case; (4) removal of emails and (5) removal of extra spaces. In addition to this, experiments were conducted involving a more rigorous cleaning pipeline.**

This comprehensive cleaning protocol involved performing stemming, removing punctuation, eliminating brackets, replacing contractions, removing stop words, removing HTML tags, removing single characters, digits and again single characters, and employing a tokeniser to prepare the text in a format suitable for BERT base models. The chosen base models for testing are prepared to manage emojis, featuring distinct representations for various emojis. Consequently, our pre-processing pipeline does not treat emojis in a special manner.

The results of the pre-processing experimentation revealed a substantial decrease in model performance when employing the heavy cleaning pipeline. This highlighted the importance of certain elements such as punctuation and common words that are typically categorised as stop words, as they appear to contribute significantly to the model's ability to classify sentiment. Furthermore, we have conducted trials with a less intensive cleaning pipeline, wherein all the aforementioned pre-processing steps were applied, except

<sup>80</sup> Farra, N., et al., 'SemEval-2017 task 4: Sentiment analysis in Twitter,' *Proceedings of the 11th International Workshop on Semantic Evaluation*, 2017.

<sup>81</sup> Ibid.

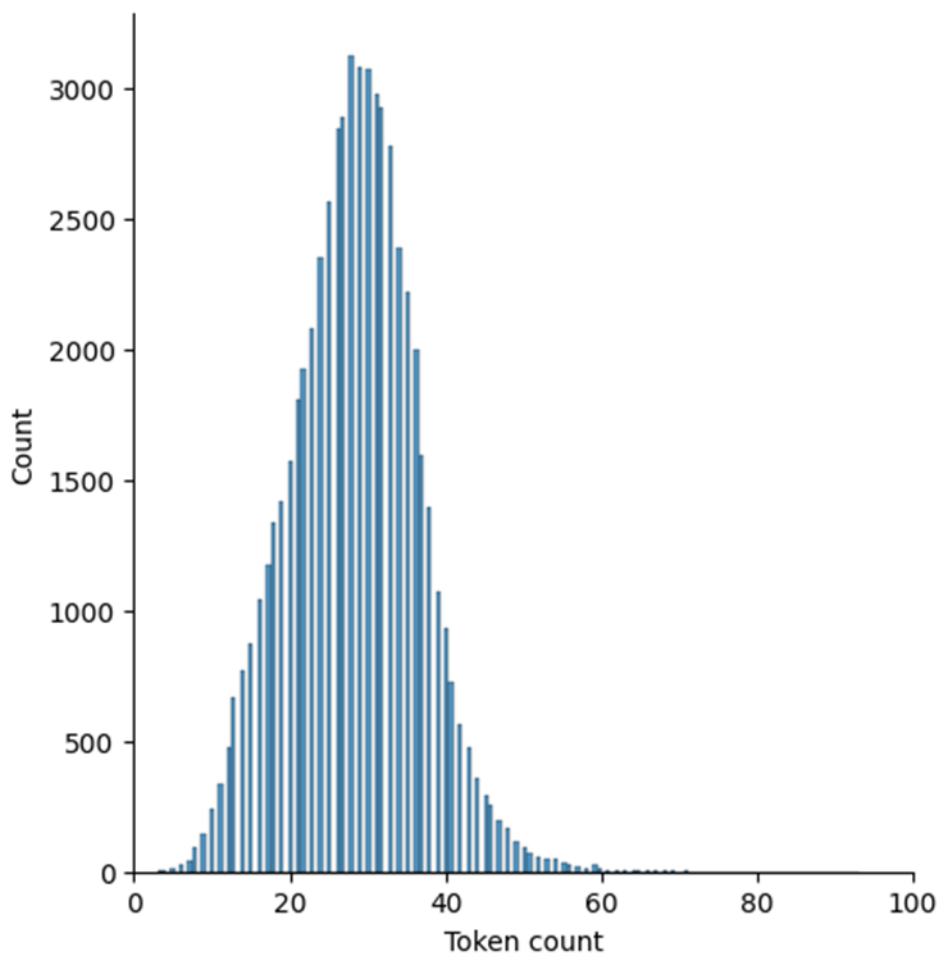
<sup>82</sup> Dang, N.C., et al., 'Sentiment Analysis Based on Deep Learning: A Comparative Study,' *Electronics*, 2020.

for stemming, punctuation removal, and the elimination of stop words. In this case too, anonymisation proceedings were still fully carried out to comply with FERMI's data protection obligations.

### 5.2.1.3 Encoding

**The encoding process for input text, within the framework of a BERT-based model, encompasses a series of pivotal stages meticulously crafted to pre-process textual data prior to its ingestion into the model.** This implemented encoding procedure entails several steps, including tokenisation, the addition of classification and separator tokens, padding of inputs to a fixed length, conversion of tokens into corresponding IDs, assignment of segment IDs to each token, generation of an attention mask distinguishing actual words (non-padding tokens) from padding tokens, the final formatting of input data for compatibility with the BERT model, and the organisation of multiple input examples into batches to optimise efficiency, during both the training and inference phases. **Each of the models subjected to experimentation was equipped with its own pretrained tokeniser**, which adeptly manages the encoding process in the manner previously elucidated.

To achieve an efficient input encoding process, **a fixed input length needed to be established.** To decide upon said input length, an analysis of the collected X posts' length was done. The majority of X posts typically range from 20 to 40 tokens, with the largest observed containing 70. In alignment with the model, which was trained on sequences of 510 tokens, a maximum length of 512 tokens was chosen. This guarantees the inclusion of complete X posts or sentences within our training dataset, thus preserving data integrity and model compatibility. Figure 24 presents the frequency of tokens contained in the X posts included in the TweetEval dataset.



**Figure 24: Frequency of tokens per tweet**

#### 5.2.1.4 Model Selection and Evaluation

For selecting the appropriate model for each step of the training phase, the following two fundamental methodologies have been examined, feature extraction and fine-tuning. Feature extraction leverages a pre-trained model whilst the fine-tuning method involves persistent modifications to the model's architecture through appending additional layers to the pre-trained model's structure.<sup>83</sup> The initial step in this research involved the identification of approaches and architectural configurations including the base model's selection, encoding, fine-tuning and feature extraction deemed worthy of examination. Within the context of this experiment, two distinct strategies were chosen and evaluated. The first architectural configuration involved the utilisation of a pretrained base model, either RoBERTa or BERT, augmented by a straightforward classifier positioned at the top of it. This top classifier underwent fine-tuning while leaving the layers of the pretrained base model trainable. Conversely, the second selected methodology adopted a BERT-based pretrained model as an adept feature extractor, proficiently capturing bidirectional patterns. This model architecture was further complemented with a LSTM classifier. The rationale behind selecting the bi-LSTM classifier stems from its inherent competence in adeptly capturing extensive and long-range dependencies within the dataset. In this latter architecture, token embeddings were fed into the LSTM classifier. The experimental procedure encompassed a comprehensive evaluation of both methods, followed by a comparative analysis to discern the most optimal choice for the Sentiment Analysis module.

In the domain of text classification, the experimental assessment involved renowned models recognised for their SOTA performance, specifically BERT, RoBERTa, DistilBERT, and ALBERT.<sup>84</sup> These models are built upon the framework of transformers, utilising an encoder to interpret text input and a decoder to generate task predictions. As BERT is primarily engineered as a language model, only the encoder mechanism is applied for the purposes of the Sentiment Analysis module. For the module, particular focus was placed on the training a model tailored for extracting sentiment polarity feature. To ensure due diligence, the proficiency of RoBERTa and BERT in sentiment analysis was thoroughly explored. In tandem with architectural considerations, the investigation into said models' proficiency delved into the training process of BERT-based models. These models, during their training regimen, meticulously analyse a designated training dataset to refine their internal "kernel parameters," which the GA expects to be used.<sup>85</sup> The primary aim of this meticulous refinement is to enhance accuracy. Within this training framework, BERT exhibits a remarkable ability to distil and extract meaningful patterns from the input text, encapsulating these insights as feature maps, which significantly contributes to a deeper comprehension of textual content. Subsequently, these feature maps traverse to the indispensable pre-classification layer, where their interpretive capacity is harnessed to make discerning decisions. One notable application of this interpretive prowess includes sentiment analysis, enabling the models to differentiate between positive, neutral, and negative sentiments with precision and rigour.

For the fine-tuning methodology, the chosen approach involved utilising a pretrained RoBERTa model initially pretrained on the SST3<sup>86</sup> dataset and subsequently fine-tuned with the **TweetEval dataset**. In the case of the feature-extraction approach, an array of models was assessed, including RoBERTa pretrained on SST3, RoBERTa base, and RoBERTa pretrained on TweetEval, among others.

##### 5.2.1.4.1 Fine Tuning Method

In the fine-tuning approach, all layers of the pretrained base model are made trainable, permitting them to engage in the learning process throughout training. Notably, the upper layers tend to specialise in the specific task being addressed. However, as the training advances, the model gradually tends to forget previously acquired knowledge. In the course of the experiments, the alternative of freezing specific layers has also been explored. The optimisation of hyperparameters was conducted using the training split of the TweetEval dataset in conjunction with the validation split of TweetEval. Hyperparameter tuning was specifically tailored to optimise the average recall metric. This metric was chosen due to its alignment with the TweetEval dataset

<sup>83</sup> Ngoc, H.L., et al., 'Fine-Tuning BERT for Sentiment Analysis of Vietnamese Reviews,' *arXiv preprint*, 2020.

<sup>84</sup> Ibid.

<sup>85</sup> 'Grant Agreement: Project 101073980 – FERMI – HORIZON-CL3-2021-FCT-01,' European Research Executive Agency, 2021.

<sup>86</sup> See <https://huggingface.co/datasets/artemis13fowl/sst-3>.

benchmarks. Furthermore, the preference for recall can be substantiated by its robustness in addressing imbalanced datasets. The hyperparameters selected for tuning encompass the Learning Rate, Weight Decay, Per Device Train Batch Size, Per Device Eval Batch Size, Num Train Epochs, and the choice of Optimiser, which is the algorithm employed to update the model’s weights during the training process.

In addition to the aforementioned procedures, the number of trainable layers was also fine-tuned. Following the selection of optimal hyperparameters, an exhaustive search was conducted to identify the most suitable seed for data sampling and model initialisation. The resultant set of best parameters is reported in Table 3 and the outcomes in Table 4.

**Table 3: Best hyperparameters for fine-tuned model**

<b>Learning Rate</b>	1.4305135307339992e-06
<b>Weight-decay</b>	5.188348810329188e-05
<b>Num-train-epochs</b>	31
<b>Optimizer</b>	adafactor
<b>Per Device Train Batch Size</b>	8
<b>Per Device Eval Batch Size</b>	12
<b>Seed</b>	42
<b>Data-seed</b>	42
<b>No frozen layer</b>	0

**Table 4: Evaluation metrics on train and validation set**

<b>Eval Accuracy</b>	<b>Eval F1-Test</b>	<b>Eval Loss</b>	<b>Eval Recall</b>	<b>Train Accuracy</b>	<b>Train F1-Test</b>	<b>Train Loss</b>	<b>Train Recall</b>
0.742	0.727	0.630	0.728	0.782	0.775	0.622	0.776

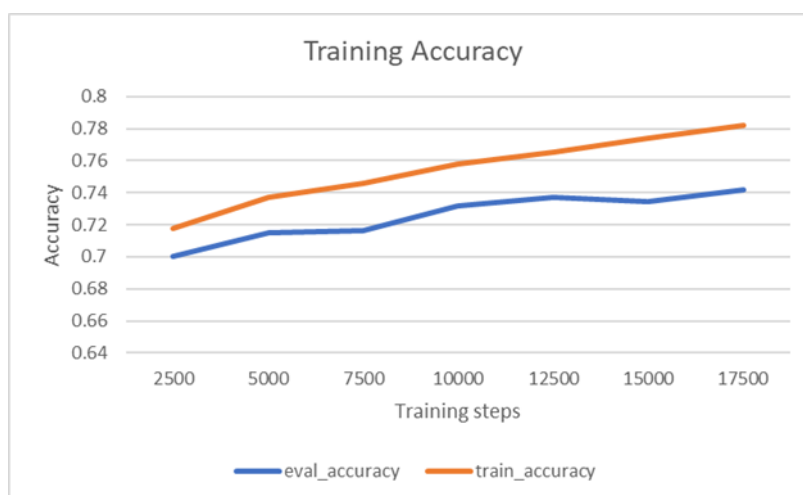
The choice of this particular model was predicated upon its robustness, as evidenced by the relatively minor disparity between its training and evaluation metrics. This implies that the model demonstrates robust generalisation capabilities. Using extracted data from the model’s lifecycle (mlFlow),<sup>87</sup> the training and evaluation loss of the chosen model across training steps is depicted in Figure 24.

<sup>87</sup> See <https://mlflow.org/>.



**Figure 24: Loss of the model during the training process**

It is evident that **the loss during training (orange line) and the loss during evaluation (blue line) do not exhibit large difference, indicating that the model has generalised well to unseen data.** Moreover, Figure 25 reinforces the robustness of our model, as evidenced by the closely aligned values of training and evaluation accuracy.



**Figure 25: Training and evaluation accuracy across training steps**

The model’s performance on the test set can be seen in Table 5.

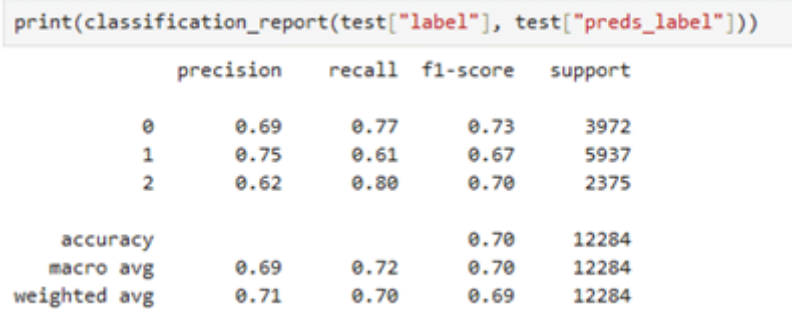
**Table 5: Evaluation metrics on test set**

Test Accuracy	F1 - Test	Test Recall	Test Precision
0.700	0.699	0.722	0.69

As previously discussed, an in-depth examination of the influence of pre-processing techniques on the model’s performance has been undertaken. In this subsection, we provide an overview of the outcomes obtained from extensive text pre-processing methods. Additionally, it is of significance to present the evaluation metrics pertaining to the model trained with optimal hyperparameters but on a dataset subjected to comprehensive data cleaning procedures (see Table 6). **Given the comprehensive analysis of the evaluation metrics, which collectively indicate a reduction in performance, the decision was made to maintain the current pre-processing pipeline in its existing form.** This entails retaining stop words, punctuation marks, and abstaining from lemmatisation processes.

**Table 6: Evaluation metrics of the model trained on a meticulously cleaned training dataset, involving the removal of stop words, punctuation marks, and lemmatisation processes**

Eval Accuracy	Eval F1-Test	Eval Loss	Eval Recall	Train Accuracy	Train F1	Train Loss	Train Recall
0.673	0.645	0.755	0.636	0.718	0.698	0.767	0.686



**Figure 26: Evaluation metrics for each class**

Metrics pertaining to each individual class for this model have been obtained and are reported in Figure 26. Notably, the *neutral* class displays the lowest accuracy and recall among all the classes, implying a higher incidence of inaccuracy when the model allocates instances to this specific class. This phenomenon can be attributed to the inherent class imbalance present in the datasets, with the neutral class being disproportionately represented across all data splits. Undoubtedly, the disproportionate representation of the neutral class within the dataset can impart a misleading impression to the model, implying a higher prevalence that may not be the case. Such a scenario, recognised as overfitting, can result in suboptimal models with limited generalisation capabilities, diminished sensitivity, substandard learning performance, and skewed evaluation metrics. In the process of model training, it becomes imperative to curate a dataset that embodies a balanced and representative distribution of data points across each class. This balanced representation is fundamental to enabling the model to acquire an accurate understanding of the distinctive attributes characterising each class. Consequently, this equilibrium in data representation reinforces the meaningfulness and robustness of the model, ensuring their applicability across the entire spectrum of classes.

In order to gauge the influence of class imbalance on the model’s performance, oversampling, weight loss functions, and supplementing the dataset (with X posts from t4sa) were applied, individually. It was imperative that the model underwent training using the identical set of optimal hyperparameters, while a rigorous performance assessment was conducted. The results presented in Table 7 pertain to each of the class imbalance mitigation approaches.

**Table 7: Evaluation metrics of the model trained on different strategies of balancing the training set.**

	Oversampling	Weight Loss Functions	Supplementation
Eval Accuracy	0.727	0.728	0.727
Eval F1-Test	0.713	0.714	0.711
Eval Loss	0.708	0.668	0.662
Eval Recall	0.743	0.735	0.72
Train Accuracy	0.835	0.767	0.82
Train F1	0.832	0.764	0.818
Train Loss	0.579	0.645	0.518
Train Recall	0.835	0.787	0.82
Test Accuracy	0.67	0.687	0.692
Test F1-Test	0.674	0.69	0.694



Test Recall	0.714	0.724	0.714
-------------	-------	-------	-------

In the first strategy, oversampling occurred within the underrepresented classes of TweetEval. In the second, weight loss functions, assigning greater weight to the underrepresented classes, were implemented, and, in the third approach, the dataset was supplemented with additional observations from the underrepresented class.

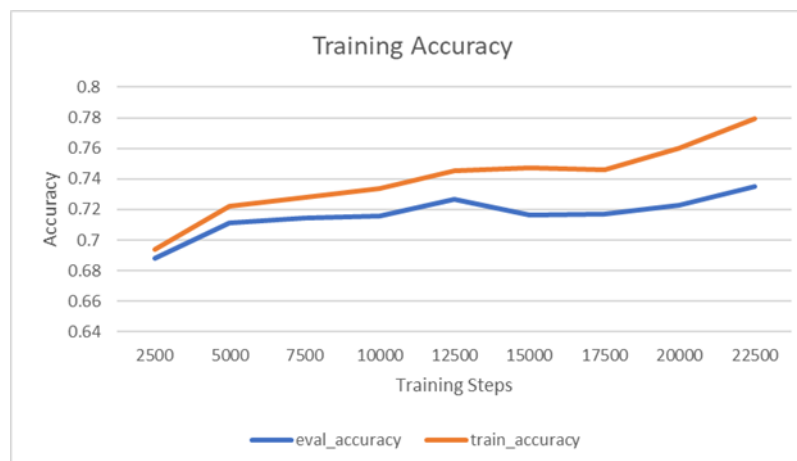
As inferred from the observations, achieving dataset balance yields a discernible, albeit moderate, improvement in performance. To further investigate this aspect, an alternative course of action involves a subsequent iteration of hyperparameter tuning. This entails an evaluation of whether distinct parameter configurations exhibit superior suitability for a balanced dataset, with the potential to yield additional performance enhancements. Due to the enhanced performance achieved through weight loss functions, it was, subsequently, decided to train this model on the integrated train and test split of TweetEval, encompassing a total of 57,899 data rows. Following this comprehensive training, validation was performed on a distinct validation split comprising 2,000 data rows. Table 8 presents the metrics acquired during the concluding stages of its training process for this model, the one that will be provided to the FERMI platform. The progress regarding loss and accuracy during the model’s training is depicted in figures 27 and 28. As both lines closely track each other, there is no apparent indication of overfitting.

**Table 8: Evaluation metrics of the model trained on the concatenated train and test dataset.**

Eval Accuracy	Eval F1-Test	Eval Loss	Eval Recall	Train Accuracy	Train	Train Loss	Train Recall
0.735	0.724	0.672	0.736	0.779	0.78	0.628	0.797



**Figure 27: Loss of the model during the training process**



**Figure 28: Accuracy of the model during the training process****5.2.1.4.2 Feature Extraction Method**

The methodology employed involved utilising a pretrained model as a feature extractor, at the top of which a classifier of choice was added, subsequently trained using the TweetEval dataset. The process of feature extraction involves utilising a BERT-based model, which transforms the pre-processed text into an attention mechanism. This mechanism learns contextual relationships between words (or sub-words) in a text or post, ultimately generating token embeddings. An LSTM model, leveraging those token embeddings from any BERT-based pretrained model obtained from Hugging Face, has been implemented for this purpose. Within this approach, the pretrained model was tested, along with token embeddings.<sup>88</sup> In this particular iteration, token embeddings have been extracted from the final layer of the feature extractor. It is worth noting that, given the utilisation of the pretrained model solely as a feature extractor, all layers remain frozen during the training process. An additional parameter subjected to tuning is the number of LSTM layers. Multiple pretrained models from Hugging Face were employed in this context (RoBERTa fine-tuned sentiment, RoBERTa base, Twitter RoBERTa base sentiment, and BERT base uncased SST2).

The hyperparameters selected for tuning, in this approach, remain consistent with the previous settings. Furthermore, architectural considerations extend to parameters associated with LSTM layers, notably their depth. Deeper models have consistently exhibited heightened efficacy in addressing intricate tasks and accommodating a broader spectrum of data patterns. However, realising the full potential of these deep architectures necessitates meticulous attention to regularisation techniques and parameter fine-tuning to mitigate concerns such as overfitting and increased computational demands. Striking an optimal equilibrium between model depth and the complexity of the task at hand becomes imperative. In the course of experimenting, assessments across various LSTM layer quantities were conducted. Findings consistently showed that the most favourable outcomes were obtained with a model that featured 6 LSTM layers. As a result, when adopting the base pretrained model, **we adhered to the stability of this 6-layer LSTM architecture**. Table 9 reports the performance metrics for each pre-trained model.

**Table 9: Evaluation metrics of feature extraction on different pretrained models**

Model	Eval Accuracy	Eval F1-Test	Eval Loss	Eval Recall
RoBERTa Fine-Tuned Sentiment (SST3)	0.683	0.657	0.701	0.658
RoBERTa Base	0.717	0.699	0.685	0.705
Twitter-RoBERTa base sentiment	0.789	0.6780	0.500	0.791
BERT Base uncased SST-2	0.685	0.672	0.764	0.681
Model	Train Accuracy	Train F1-Test	Train Loss	Train Recall
RoBERTa Fine-Tuned Sentiment (SST3)	0.691	0.691	0.7694	0.689
RoBERTa Base	0.745	0.733	0.680	0.734
Twitter-RoBERTa base sentiment	0.815	0.811	0.494	0.822
BERT Base uncased SST-2	0.822	0.815	0.621	0.824
Model	Test Accuracy	Test F1-Test	Test Loss	Test Recall
RoBERTa Fine-Tuned Sentiment (SST3)	0.657	0.657	0.762	0.677
RoBERTa Base	0.671	0.673	0.710	0.697
Twitter-RoBERTa base sentiment	0.713	0.715	0.675	0.730
BERT Base uncased SST-2	0.673	0.671	0.727	0.678

<sup>88</sup> Chang, M., 'BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding,' *arXiv preprint*, 2018.

From Table 9, it is evident that the results of all the pre-trained models, with the exception of the Twitter RoBERTa base sentiment, exhibit accuracy and recall values of approximately 70%. Twitter RoBERTa base sentiment is notably superior, likely due to its pretraining on the TweetEval dataset. Despite the notably higher metrics of this model, the substantial disparity between the metrics on the training and test sets indicate the presence of overfitting.

Thus, **the RoBERTa base was chosen as the base model** for the purposes of the Sentiment Analysis module, as it not only **demonstrates good performance in terms of accuracy and recall, on the validation set, but also exhibits robustness, with closely aligned metrics on both the training and test sets.** Table 10 reports its ideal hyperparameters. Table 11, subsequently, presents the best-performing model's results on the test set.

**Table 10: Ideal hyperparameters for RoBERTa base**

<b>Learning-Rate</b>	9.687457389826624e-05
<b>Weight-Decay</b>	7.75376739624082e-05
<b>Num-Train-Epochs</b>	82
<b>Optimiser</b>	adafactor
<b>Per Device Train Batch Size</b>	12
<b>Per Device Eval Batch Size</b>	12
<b>Seed</b>	42
<b>Data-Seed</b>	42

**Table 11: Evaluation metrics of feature extraction model on test set.**

<b>Test Accuracy</b>	<b>F1-Test</b>	<b>Recall Test</b>
0.696	0.694	0.698

### 5.2.1.5 Training Phase Result

The research carried out reveals that the **fine-tuning approach yielded the most promising results**, with the highest average recall metric. The Sentiment Analysis module is aligned with known benchmarks, **with an average recall metric of 72.2%** when assessed on the TweetEval test dataset. Subsequent investigations, into various pre-processing pipelines, indicate, surprisingly, that actions such as stop word and punctuation removal, as well as lemmatisation, led only to marginal decreases in the evaluation metrics.

Additionally, the class imbalance issue was addressed using three distinct strategies within the training dataset. These strategies encompassed in-dataset oversampling, the utilisation of weight loss techniques to assign higher weights to underrepresented classes, and the execution of oversampling using an external t4sa dataset that contained categorised X posts with positive, negative, and neutral sentiments. Interestingly, our results demonstrated that the most substantial improvement in evaluation metrics was achieved through the second approach, which involved the utilisation of weight loss techniques. This latter **model achieved an average recall metric of 72.4%** when evaluated on the TweetEval test dataset.

It is important to note that the training process can be repeated in the future if high-quality, validated, and annotated data more aligned with the end-user's needs become available. This procedure would ensure that the model remains current and consistently relevant to the specific use case.

### 5.2.2 Inference Phase

In the inference phase, a series of pipelines were established to manage data ingestion and preparation, streamlining the trained model serving process and enabling output generation. The input data source emanates from the Spread Analyser, represented as a data graph structure. For each node within this graph, the text content of the corresponding retrieved tweet is extracted. Subsequently, a sequence of pre-processing and

---

encoding steps is applied before feeding the data into the model. **The resulting output maintains the graph structure, but now each node incorporates additional attributes: the predicted sentiment label and the associated accuracy probability.**

To facilitate this process, **a REST API was developed** using the FastAPI tool.<sup>89</sup> In the process described, the API takes in the entirety of a graph structure as its input, initialises the model, and systematically shepherds each node within the graph through a sequence of procedural steps. These steps involve initial pre-processing, subsequent encoding, and culminate in the passage of each node through the model, yielding a sentiment estimate. Consequently, the output maintains the initial graph structure but undergoes further processing, which includes the incorporation of supplementary attributes associated with sentiment analysis, all in strict compliance with the pertinent project's ethics guidelines.

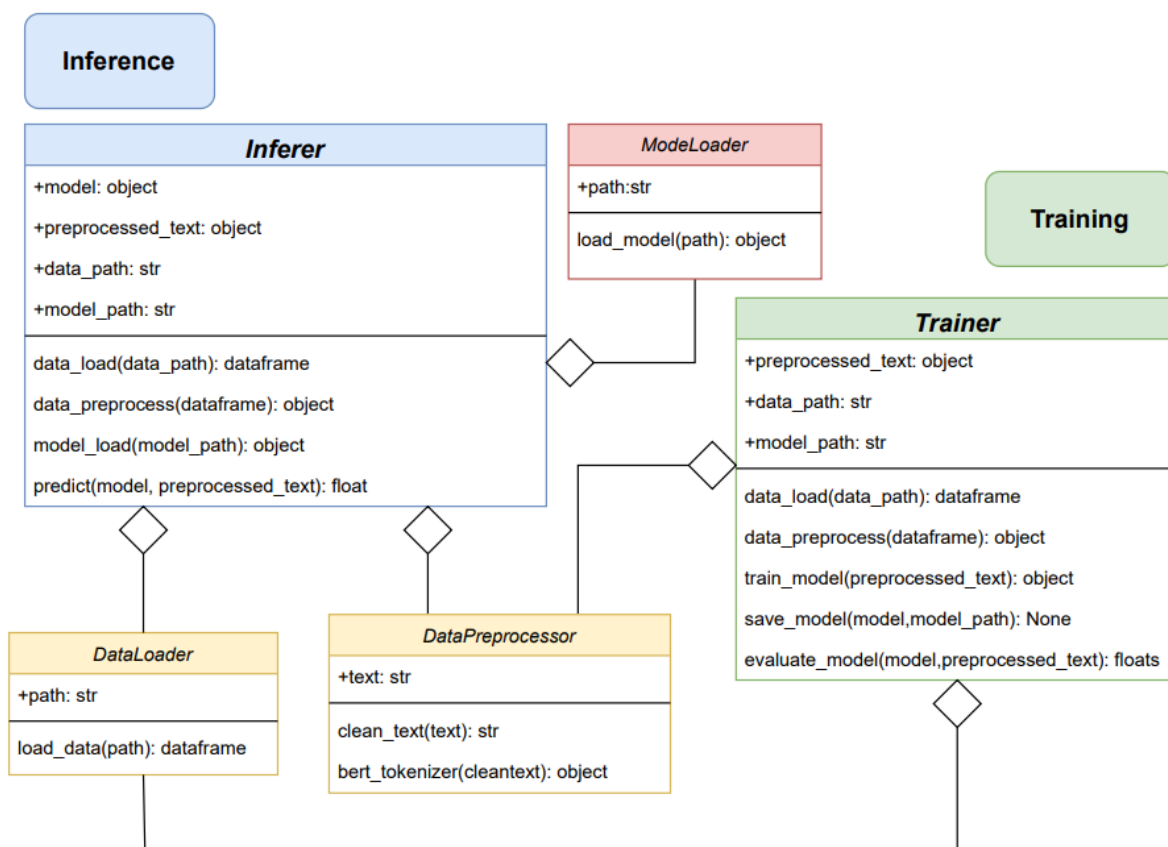
The Sentiment Analysis module class structure is illustrated in Figure 29, comprised of two primary classes that fulfil distinct roles within the system. The first is an *inferer* which handles inference, whilst the second class is a *trainer*, that focuses on training. Subclasses complement the aforementioned, including the *data-loader* and *data-processor*, which have an aggregated relationship with the main classes as well as the *model-loader*. The inferer class employs data to conduct sentiment analysis based on the trained model, a process that involves the subclasses. The data-loader retrieves the data from the platform, the data-processor cleans and tokenises said data, and the model-loader creates the model object utilised by the inferer class for its sentiment estimates.

The output of the Sentiment Analysis module is derived from estimates generated by the RoBERTa model. It provides sentiment labels, including positive, negative, or neutral, along with sentiment scores or their combined representations. These outputs encompass a broad spectrum, ranging from overall sentiment scores for analysed data graphs to sentiment scores over time and even sentiment scores per graph node.

---

<sup>89</sup> 'FastAPI,' *FastAPI*, n.d.

DataLoader, DataPreprocessor and ModelLoader are parts of Inferer, but they can exist on their own



**Figure 29: Sentiment Analysis module class diagram**

### 5.2.3 Challenges and Limitations

#### 5.2.3.1 Specificity Versus Accuracy

Training a model is highly dependent on the dataset utilised in the process. The closer the training dataset aligns with the target data, the more precise the model's predictions will be. This presents a limitation for the model's performance across **different social media platforms** and **varied domains**.

Social media channels pose a unique set of challenges for NLP, primarily attributable to several distinctive characteristics inherent to each platform. One said challenge stems from the brevity of each social media's posts, necessitating the use of concise and novel language specific to the channel<sup>90</sup>. This brief mode of communication often incorporates slang and acronyms and is further constrained by the platform's character limit. Consequently, social media users develop a rapidly evolving and unique vocabulary, which is differentiated across platforms, that presents a formidable challenge for analysis. In summary, these factors collectively underscore the complexities associated with analysing and interpreting content on social media platforms and emphasise the limitations of developing a highly accurate generalised model for all social media channels.

Additionally, models trained on domain general data often exhibit lower evaluation metrics compared to those trained on domain-specific data due to the inherent complexity and diversity of general datasets<sup>91</sup>.

<sup>90</sup> Giachanou, Anastasia & Crestani, Fabio, 'Like It or Not: A Survey of Twitter Sentiment Analysis Methods,' *ACM Computing Surveys*, Vol. 49, 2016, pp. 1 – 41.

<sup>91</sup> Lim, C.Y., Tan, I.K.T., Selvaretnam, B. (2019). Domain-General Versus Domain-Specific Named Entity Recognition: A Case Study Using TEXT. In: Chamchong, R., Wong, K. (eds) *Multi-disciplinary Trends in Artificial*

Domain general data encompasses a wide range of topics, styles, and contexts, making it more challenging for models to capture nuanced patterns and specialised knowledge. In contrast, domain-specific data is tailored to a particular subject or industry, allowing models to focus and specialise, thereby achieving higher performance within the specific domain. The broader nature of general data introduces noise and variability that may hinder the model's ability to generalise effectively, resulting in comparatively lower evaluation metrics.

### 5.2.3.2 3 – Class Sentiment Analysis

**The decision was made to adopt a 3 – class approach**, as opposed to a 2 – class approach, as it introduces inherent complexity to the model. It necessitates distinguishing between three distinct sentiment categories, presenting models with a more intricate task. Moreover, datasets for 3 – class sentiment analysis frequently exhibit imbalance, an issue faced in the development of the Sentiment Analysis module, with the neutral class containing a notably larger number of instances than the positive and negative classes. This imbalance can lead to model bias towards the majority class (neutral), potentially resulting in diminished accuracy for the minority classes (positive and negative). Semantic ambiguity and challenges with respect to labelling observations are also addressed with a 3 – class approach. Just as well, since the choice of evaluation metrics also can significantly influence perceived performance, as, with a 2-class analysis, achieving a high accuracy level is more easily achieved as compared to a 3-class analysis. Thus, using a 3-class analysis, and achieving a high accuracy level, implies a more robust sentiment analysis technology has been developed.

That being said, depending on our advancements, we might give **end -users the option to choose between two distinct models: one equipped for 3-class sentiment analysis and the other for 2-class sentiment analysis**. The 2 – class model, differentiating between positive and negative sentiments, holds several advantages for users in terms of rendering sentiment interpretation more straightforward, facilitating precise sentiment discernment, streamlining the analytical process, bolstering decision-making capabilities, furnishing focused insights, enhancing operational efficiency, and ensuring a heightened level of result reliability. This strategic realignment empowers users to swiftly grasp the emotional nuances within X posts, particularly when scrutinising content pertaining to D&FN, thereby enabling more assured and informed sentiment assessments and impact evaluations.

## 5.3 Current Advancement and Demo

According to the GA, the Sentiment Analysis module must be capable of being deployed to provide further analysis of D&FN, precisely “carrying out sentiment analyses to posts from social media accounts.”<sup>92</sup> Moreover, there was a commitment made to “exploit the BERT model... with a wide variety of NLP tasks.”<sup>93</sup> **As it stands, the Sentiment Analysis module employs the RoBERTa base model, achieving an average recall metric of 72.4%** on the TweetEval test set. Furthermore, using the benchmarks, provided by the TweetEval, provides verification of the accuracy of outcomes, as desired by the GA.

## 5.4 Next Steps

The proceeding steps in developing the Sentiment Analysis module are focused on refining its accuracy. The initial optimisation endeavours will involve a deliberate exploration and, if decided, transition to a binary classification system, featuring two classes, with the primary objective of streamlining sentiment categorisation for heightened precision. In parallel, active exploration is underway to assess the feasibility of implementing sentiment analysis in multilingual contexts, driven by the recognition of the importance of discerning nuanced sentiments across diverse linguistic landscapes.

Furthermore, this methodological approach encompasses a dedicated effort to enhance the existing 3-class model, concentrating on performance improvement through the systematic exploration and integration of supplementary datasets more suitable for the project's use cases. In addition, we plan to explore additional datasets from other social media platforms with sentiment labels to develop a more generalised model.

---

Intelligence. MIWAI 2019. Lecture Notes in Computer Science, vol 11909. Springer, Cham.  
[https://doi.org/10.1007/978-3-030-33709-4\\_21](https://doi.org/10.1007/978-3-030-33709-4_21)

<sup>92</sup> ‘Grant Agreement: Project 101073980 – FERMI – HORIZON-CL3-2021-FCT-01,’ *European Research Executive Agency*, 2021.

<sup>93</sup> Ibid.

---

However, it is important to note that this generalisation may come with a trade-off in accuracy, which we will also evaluate. These strategic initiatives collectively aim to elevate the precision, scope, and robustness of the sentiment analysis framework employed by the research team.

In response to the multiclass challenge, an additional model singularly dedicated to the discernment of positive and negative sentiment is strategically poised for launch. The confidence behind this endeavour stems from the anticipation that, through the meticulous fine-tuning of a pretrained model using a curated dataset comprising social media posts distinctly classified as either positive or negative, **a model exhibiting an accuracy surpassing the 90% threshold can be obtained**, as required by the GA that stipulates that “sentiment analyses with >90% accuracy” is to be achieved.<sup>94</sup>

In subsequent stages, there exists a clear ambition to conduct validation of the ML algorithms employed in the preliminary iteration of the Sentiment Analysis module. This validation will be accomplished through real-life experiments and the utilisation of data sourced from the pilot initiatives. The overarching objective of this effort is to achieve an enhanced level of intelligence within the FERMI offerings and to propel the module’s TRL to TRL-6 as per ITML’s 3ACEs toolkit,<sup>95</sup> where the Sentiment Analysis module assumes its role. Following this phase, the systematic incorporation of user feedback will be done and fine-tune the machine learning algorithms conducted, drawing insights from real-world experiments conducted within the FERMI framework. This iterative approach aims not only to augment the module’s capabilities but also to secure its resilient performance across varied scenarios, in alignment with the principles outlined in ITML’s 3ACEs toolkit. These advancements highlight our commitment to excellence and our continuous effort to develop innovative solutions in the field.

---

<sup>94</sup> Ibid.

<sup>95</sup> The 3ACEs toolkit can be at <https://www.itml.gr/products/analytics-as-a-service>.

## 6 Integration with Tasks 3.3 and 3.5

**Tasks 3.3 and T3.5 operate within the greater FERMI platform and rely on the flow of data analysis that begins with the Spread Analyser.** However, most important to their operation is the Dynamic Flows Modeler, whose output is used as the input for T3.5. The aim of these two tasks, the Behaviour Profiler & Socioeconomic Analyser and the Community Resilience Management Modeler focuses on assessing the implications of the end-user provided D&FN in terms of severity and likelihood and providing potential countermeasures the end-user can employ. In section 6, the technologies will be briefly explained, with a majority of the attention placed on the integration between them and the technologies central to D3.1 (i.e., T3.1, T3.2, T3.4, T3.6). For greater coverage of T3.3 and T3.5’s development and adherence to the GA, one should refer to D3.3: FERMI Behavioural Analyses and Community Resilience Facilitators Package - 1st version.

### 6.1 Task 3.5 – The Behaviour Profiler and Socioeconomic Analyser

The Behaviour Profiler consists of two sub-tracks, the necessity to understand the number of offline crime occurrences that may be D&FN-induced and/or -enabled following an online D&FN campaign, and an analysis of how the resident of a specific country may react to the same online D&FN campaign, considering factors including media literacy and information consumption behaviour. More specifically, the Behaviour Profiler, in its role as an impact analyser, requires the necessity to foresee the number of offline crime occurrences, which is provided by the Dynamic Flows Modeler. Using past crime and past D&FN, the Dynamic Flows Modeler estimates the change in crime occurrences for several crime types, in the end-user’s requested NUTS2 region.

As described in the GA, the Behaviour Profiler, specifically its *country profiles*, relates to “politically motivated extremism[‘s] ... impact on society... [with an aim to] determine effects of online propaganda on offline actions. In this respect, the degree of media literacy may tend to correspond to the degree of resilience of a society. The means of information and news consumption is a first indicator for the assessment of media literacy. Factors such as the type of source, the ‘general’ assessment of the medium, the level of trust (if feasible) and differentiation by age groups (demographics) play an important role. Based on secondary literature, an analysis of the media literacy of certain countries will be conducted, considering the factors mentioned above. This preliminary work allows behavioural profiles to be better differentiated and classified.”<sup>96</sup> Working towards this direction, the analysis conducted places a particular focus on media literacy in determining the effects of online D&FN on offline actions. Based on secondary literature, the Behaviour Profiler examines if, and how, media literacy in specific EU countries affects the spreading of D&FN. The country profiles chosen were the 5 target countries relevant to/for FERMI LEAs: Finland, Sweden, Belgium, Germany, and France.

As for the **Socioeconomic Analyser**, it is concerned with the connection between D&FN and crime, specifically said crime’s effects on economically measurable variables, such as GDP per inhabitant. According to the GA, the Socioeconomic Analyser, through “applying econometric methods” will reveal “the effects of radicalization and extremism... reflected in financial terms to quantify the costs of... [D&FN’s] negative effects (based on data availability in the respective country/region) for the society.”<sup>97</sup> **The intuition behind this originates from academic research** that built the theoretical foundation for looking at possible effects of violent political extremism on economic variables. It has been proposed that **violent political extremism leads to a loss in social welfare** via different channels. These may concern the deterrence of investors, the influence on political decisions and instructional output as well as trade and further factors.<sup>98</sup>

The main model to explain economic costs by politically motivated crime is depicted in the following regression equation.

$$Prod_{rt} = \alpha + \beta_1 Ext + X_{rt} + v_r + \varepsilon_{rt}$$

**Equation 12: The calculation of economic costs to political extremism**

<sup>96</sup> Ibid.

<sup>97</sup> Ibid.

<sup>98</sup> Ferguson, N., et al., ‘Die Kosten des Extremismus,’ *BIGS Standpunkt zivile Sicherheit*, Vol. 9, 2019.



It explains the measurement of productivity for a given region ( $r$ ) for a given time period ( $t$ ).  $\alpha$  is the constant for the regression,  $v_r$  is a vector of time-invariant region-specific properties.  $\varepsilon_{rt}$  is the error term for the regression. ***Ext* is the measurement of extremism in terms of crime** and  $\beta_I$  gives the cost coefficient, i.e. how a single unit increase in crime will affect economic welfare. Lastly,  $X_{rt}$  describes a vector of control variables, such as size of the region.

It is in this equation that the integration between technologies is most relevant. The number of ***Ext*, in this equation, is sourced from the Dynamic Flows Modeler**, which provides a level of offline crime occurrence following a D&FN event online. Recalling that the past D&FN events studied by the Dynamic Flows Modeler correspond to political extremism, at its current state of development, particularly right-wing extremism. Figure 30 presents the data flow between FERMI components, as a whole. Within said figure, the linkage between the Dynamic Flows Modeler (D&FN Offline Crime Analyses) and the Behaviour Profiler & Socioeconomic Analyser can be seen. Just as well, the outputs of the Behaviour Profiler & Socioeconomic Analyser are then passed to the Community Resilience Management Modeler.

## 6.2 Task 3.3 – The Community Resilience Management Module

The Community Resilience Management Modeler and Disinformation watch joint component aims to aid LEAs in prioritising the correct course of action for tackling D&FN-related crime. Considering the impact produced by a crime, the tool will output a ranking of countermeasures to tackle high-stakes D&FN events. This output results from a multi-criteria decision analysis that produces a decision model and has the LEA as the decision-maker. The decision model and subsequent additive model will consider the LEA consensual opinion to provide options for tackling disinformation. Achieving a consensus on what options to adopt will be obtained through a DELPHI study initiative. Furthermore, the decision model will consider predefined criteria, referring to a specific factor the decision-maker uses to evaluate and assess alternatives under consideration (e.g., media literacy index, media trust index, thread of poverty or social exclusion, etc.).

The component will also consider a previously made assessment of a D&FN event's impact, measured by the product of the likelihood and the severity of a particular crime occurring in a specific community. Provided that the assessment has an index value indicating that an investigation is of high or extremely high impact on the community, the system will output a ranking of countermeasures specific to the instance of crime being investigated by the end-user. On the other hand, should the impact index be minimal or medium, the system will not output any countermeasures, and a message that no action is advisable will be provided.

In this way, LEAs will gain insight into whether to reallocate resources to counteract the D&FN under investigation or not. The impact assessment of a D&FN event on the community in question is provided by the Behaviour Profiler & Socio-economic Analyser, which, once given the potential number of crime occurrences by the Dynamic Flows Modeler, generates a measure of the D&FN in question's impact, in economic terms. The flow of data between components can be seen in Figure 30.

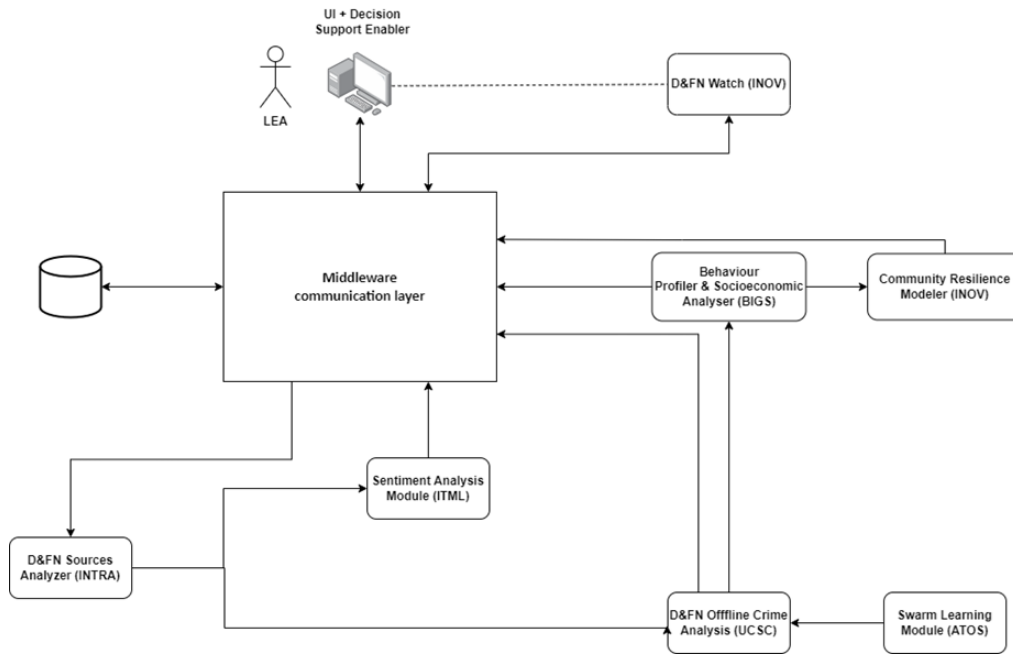


Figure 30: FERMI platform’s data flow

## 7 Conclusion

D3.1, the technology facilitators package – 1<sup>st</sup> version, offered a comprehensive review of the technologies created for the FERMI platform, importantly, it also underscores how said technological components are in compliance with the GA. Specifically, D3.1 reported on how T3.1, T3.2, T3.4, and T3.6 were developed, how they function, and remarks on their performance. The next steps for each of the technologies were also covered, in how they plan to further align with the GA and enhance the quality of the product delivered to end-users. The four technological components covered were: (T3.1) D&FN-induced and D&FN-enabled offline crimes analysis, referred to as the Dynamic Flows Modeler; (T3.2) Disinformation Sources and Spread Analysis and Impact Assessment, referred to as the Spread Analyser; (T3.4) swarm learning, for holistic AI-based services in LEA, and (T3.6) the sentiment analysis module, treated as a proper noun, that is, the Sentiment Analysis module.

The Dynamic Flows Modeler successfully evaluates the “the intensity of the relation between the spread of D&FN and offline crimes, the temporal patterns in the relation, [and] the spatial decay of the relation”<sup>99</sup> in its capacity to produce informed, accurate estimates for offline crime occurrences following a D&FN event online. The Dynamic Flows Modeler is supported by the completion of a first, functional swarm learning infrastructure that allows “for training Machine Learning models near to the data sources where they are generated,”<sup>100</sup> where the data sources are several, independent European LEAs. Given the swarm learning infrastructure, said LEAs do not need to sacrifice any degree of data privacy nor having to turn over any confidential information. Moreover, as articulated in section 3, the Spread Analyser is a powerful technology capable of taking “as input news already classified as [D&FN] and... [mapping] this news to their main actors/accounts which are responsible for creating and spreading the [D&FN] across the network.”<sup>101</sup> The Spread Analyser, adhering to the GA’s commitments, can classify if the identified actors/accounts are physical persons or bots and assign an influence index to their role/power over the network. The Sentiment Analysis module, which analyses D&FN, specifically in social media posts, to provide end-users a perception of the emotional tone in said posts’ content, exploits BERT, as promised in the GA. Ensuring the anonymisation of the posts, deletion of links, and replacing of emoji characters with corresponding text/keyword, the Sentiment Analysis module provides end-users with a wholistic understanding of the sentiment behind the network sharing the D&FN they are investigating.

T3.3 and T3.5, further offerings by the FERMI platform, are examined in depth in D3.3, however, D3.1 mentions, with brevity, their functionality and how they are integrated with the technologies in focus. Specifically, how the Dynamic Flows Modeler provides the estimated number of crime occurrences, from which the Behaviour Profiler and Socioeconomic Analyser produce an understanding of the potential harm, in economic terms, deriving from the D&FN provided by the end-user. The Community Resilience Management Modeler, then, informs end-users as to what potential counter measures may be taken.

These tools represent significant advancements, when compared to the SOTA technologies available to LEAs who are seeking means by which they can understand the threat of D&FN-induced crimes, as well as tension, and assess the risk they may represent to society. They provide needed insights, with which LEAs can make better-informed choices regarding the distribution of resources, practically given the pre-FERMI fog around the depth of a network behind a given D&FN post, a fog the Spread Analyser pushes aside. Moreover, FERMI now provides LEAs with a more objective and accelerated assessment of the sentiments held by those interacting with D&FN, as well as an AI-driven estimate of various offline crime types, in the weeks that follow a D&FN event online. An estimate reached by having ML models study confidential LEA data that, thanks to the swarm learning infrastructure, needs not be shared.

<sup>99</sup> ‘Grant Agreement: Project 101073980 – FERMI – HORIZON-CL3-2021-FCT-01,’ *European Research Executive Agency*, 2021.

<sup>100</sup> *Ibid.*

<sup>101</sup> *Ibid.*

## References

- Abbasi, A., Chen, H., Zeng, D., & Zimbra, D., 'The State-of-the-Art in Twitter Sentiment Analysis: A Review and Benchmark Evaluation,' *ACM Transactions on Management Information Systems*, Vol. 9, No 2, 2018, pp. 1 – 29.
- Abdeljaber, O., Avci, O., Gabbouj, M., Ince, T., Inman, D.J., & Kiranyaz, S., '1D Convolutional Neural Networks and Applications: a Survey,' *Mechanical Systems and Signal Processing*, Vol. 151, No 107398, 2021.
- Alonso, M.A., Gomez-Rodriguez, C., Vilares, D., & Vilares, J., 'Sentiment Analysis for Fake News Detection,' *Electronics*, Vol. 10, No 11, 2021.
- Amini, M., Akbari, Y., Godarzi, J.A., & Sharifani, K., 'Operating Machine Learning Across Natural Language Processing Techniques for Improvement of Fabricating News Models' *International Journal of Science System Research*, Vol. 12, No 9, 2022, pp. 25 – 44.
- Arcas, B.A., McMahan, H.B., Moore, E., Ramadge, P.J., & Ramage, D., 'Communication-efficient learning of deep networks from decentralized data'. *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, Vol. 54, 2017, pp. 1273 – 1282.
- Atiya, A.F., 'Why does Forecast Combination Work So Well?' *International Journal of Forecasting*, Vol. 36, No 1, 2020, pp. 197 – 200.
- Baly, R., Da San Martino, G., Glass, J., & Nakov, P., 'We can Detect your Bias: Predicting the Political Ideology of News Articles,' *arXiv preprint*, arXiv: 2010.05338, 2020.
- Balas, V.E., Mastorakis, N.E., Popescu, M.C., & Perescu-Popescu, L., 'Multilayer perceptron and neural networks,' *WSEAS Transactions on Circuits and Systems*, Vol. 8, No. 7, 2009, pp. 579 – 588.
- Barbieri, F., & Neves, L., 'TweetEval: Unified Benchmark and Comparative Evaluation for Tweet Classification,' *arXiv preprint*, arXiv: 2010.12421, 2020.
- Belda, S., Dhar, S., Ferrandiz, J.M., Guessoum, S., Heinkelmann, R., Modiri, S., Raut, S., & Schuh, H., 'The Short-Term Prediction of Length of Day Using 1D Convolutional Neural Networks (1D CNN),' *Sensors*, Vol. 22, No 9517, 2022.
- Bondi, A.B., 'Characteristics of scalability and their impact on performance', *Proceedings of the 2nd international workshop on Software and performance*, 2000, pp. 195 – 203.
- Botticher, A., 'Towards Academic Consensus Definitions of Radicalism and Extremism' *Perspective Terror*, Vol. 11, No 4, 2017, pp. 73 – 77.
- Bricken, A., 'Does BERT Need Clean Data? Part 1: Data Cleaning,' *Medium*, 2021.
- Burnap, P., Javed, A., Liu, H., Ozalp, S., Williams, M.L., 'Hate in the Machine: Anti-Black and Anti-Muslim Social Media Posts as Predictors of Offline Racially and Religiously Aggravated Crime,' *British Journal of Criminology*, Vol. 60, No 1, 2020, pp. 93 – 117.
- Burnap, P., & Williams, M.L., 'Cyberhate on Social Media in the Aftermath of Woolwich: a Case Study in Computational Criminology and Big Data,' *British Journal of Criminology*, Vol. 56, No 2, 2016, pp. 211 – 238.

- Carrara, F., Cimino, A., Cresci, S., Dell'Orletta, F., Falchi, F., Tesconi, M., & Vadicamo, L., 'Cross-Media Learning for Image Sentiment Analysis in the Wild,' *2017 IEEE International Conference on Computer Vision Workshops*, 2017, pp. 308 – 317.
- Chang, M., Devlin, J., Lee, K., & Toutanova, K., 'BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding,' *arXiv preprint*, arXiv: 1810.04805, 2018.
- Chapman, L., Petutschnig, A., Resch, B., Roberts, H., & Zimmer, S., 'Investigating the Emotional Responses of Individuals to Urban Green Space Using Twitter Data: A Critical Comparison of Three Different Methods of Sentiment Analysis,' *Urban Planning*, Vol. 3, No. 1, 2018, pp. 21 – 33.
- Chen, M., Xu, Q., Zeng, A., & Zhang, L., 'Are Transformers Effective for Time Series Forecasting?' *arXiv preprint*, arXiv: 2205.13504, 2022.
- Chen, S., Li, H., Pang, L., & Wen, D., 'The Relationship Between Social Media Use and Negative Emotions Among Chinese Medical College Students: The Mediating Role of Fear of Missing Out and the Moderating Role of Resilience,' *Psychology Research and Behavior Management*, Vol. 16, 2023, pp. 2755 – 2766.
- Chen, T., Liu, Y., Tong, Y., & Yang, Q., 'Federated machine learning: Concept and applications,' *ACM Transactions on Intelligent Systems and Technology*, Vol. 10, No 2, 2019, pp. 1 – 19.
- Choi, E., Choi, Y., Gabriel, S., Hallinan, S., Nguyen, P., Roesner, F., & Sap, M., 'Misinfo Reaction Frames: Reasoning about Readers' Reactions to News Headlines,' *arXiv preprint*, arXiv: 2104.08790, 2021.
- Dang, N.C., De la Prieta, F., & Moreno-Garcia, M.N., 'Sentiment Analysis Based on Deep Learning: A Comparative Study,' *Electronics*, Vol. 9, No. 3, 2020, pp. 483 – 512.
- Das, A., Kovatchev, V., Lease, M., & Liu, H., 'The State of Human-Centred NLP technology for Fact-Checking,' *Information Processing & Management*, Vol. 60, No 2, 2023.
- Ding, C., & Raza, S., 'Fake News Detection Based on News Content and Social Contexts: a Transformer-Based Approach,' *International Journal of Data Science and Analytics*, Vol. 13, 2022, pp. 335 – 362.
- Farra, N., Nakov, P., & Rosenthal, S., 'SemEval-2017 task 4: Sentiment analysis in Twitter,' *Proceedings of the 11th International Workshop on Semantic Evaluation*, 2017, pp. 502 – 518.
- 'FastAPI,' *FastAPI*, n.d.
- Ferguson, N., Rieckmann, J., & Stuchtey, T.H., 'Die Kosten des Extremismus,' *BIGS Standpunkt zivile Sicherheit*, Vol. 9, 2019.
- Ferrara, E., & Kudugunta, S., 'Deep Neural Networks for Bot Detection,' *Information Sciences*, Vol. 467, 2018, pp. 312 – 322.
- Florian Haupt, D.K., 'A Model-Driven Approach for REST Compliant Services,' *2014 IEEE International Conference on Web Services*, 2014.
- Freeman, L.C., 'A Set of Measures of Centrality Based on Betweenness,' *Sociometry*, Vol. 40, No. 1, 1977, pp. 35 – 41.
- General Project Review Consolidated Report.

- Gallacher, J.D., Heerdink, M.W., & Hewstone, M., ‘Online Engagement Between Opposing Political Protest Groups via Social Media is Linked to Physical Violence of Offline Encounters,’ *Social Media + Society*, January-March, 2021, pp. 1 – 16.
- ‘Getting Started,’ *X Developer Platform*, n.d.
- Giachanou, Anastasia & Crestani, Fabio, ‘Like It or Not: A Survey of Twitter Sentiment Analysis Methods,’ *ACM Computing Surveys*, Vol. 49, 2016, pp. 1 – 41.
- Goldreich, O., *Foundations of Cryptography: Volume 2, Basic Applications*, Cambridge University Press, Cambridge, 2004.
- Gomez, A.N., Jones, L., Kaiser, L., Parmar, N., Polosukhin, I., Shazeer, N., Uszkoreit, J., & Vaswani, A., ‘Attention is all You Need,’ *arXiv preprint*, arXiv: 1706.03762, 2017.
- ‘Grant Agreement: Project 101073980 – FERMI – HORIZON-CL3-2021-FCT-01,’ *European Research Executive Agency*, 2021.
- Gruppi, M., Adali, S., & Horne, B.D., ‘NELA-GT-2022: a Large Multi-Labelled News Dataset for the Study of Misinformation in News Articles,’ *arXiv preprint*, arXiv: 2203.05659, 2023.
- Gruppi, M., Adali, S., & Horne, B.D., ‘NELA-GT-2019: a Large Multi-Labelled News Dataset for the Study of Misinformation in News Articles,’ *arXiv preprint*, arXiv:2003.08444, 2020.
- Gruppi, M., Adali, S., & Horne, B.D., ‘NELA-GT-2020: a Large Multi-Labelled News Dataset for the Study of Misinformation in News Articles,’ *arXiv preprint*, arXiv:2102.04567, 2021.
- Gruppi, M., Adali, S., & Horne, B.D., ‘NELA-GT-2021: a Large Multi-Labelled News Dataset for the Study of Misinformation in News Articles,’ *arXiv preprint*, arXiv: 2203.05659, 2022.
- Hayes, A., ‘Autoregressive Integrated Moving Average (ARIMA) Prediction Model,’ *Investopedia*, 2023.
- Haupt, F., ‘A Model-Driven Approach for REST Compliant Services,’ *2014 IEEE International Conference on Web Services*, 2014.
- Kotamraju, S., ‘Everything you Need to Know about ALBERT, RoBERTa, and DistilBERT,’ *Towards Data Science*, 2022.
- Lim, C.Y., Selvaretnam, B., & Tan, J.K.T., ‘Domain-General Versus Domain-Specific Named Entity Recognition: A Case Study Using TEXT,’ in Chamchong, R., & Wong, K., (eds) ‘Multi-disciplinary Trends in Artificial Intelligence,’ *Lecture Notes in Computer Science*, Vol. 11909, 2019.
- Mira, J., et al. ‘D3.1 Federated Learning implementation’, *ALCHIMIA Horizon Europe Project*, 2023.
- Murali, V., ‘Everything you Need to Know about Ensemble Learning,’ *Medium*, 2021.
- Muller, K., & Schwarz, C., ‘Fanning the Flames of Hate: Social Media and Hate Crime’ *Journal of European Economic Association*, Vol. 19, No 4, 2021, pp. 2131 – 2167.
- Muller, K., & Schwarz, C., ‘From Hashtag to Hate Crime: Twitter and Anti-Minority Sentiment,’ *SSRN Working Paper*, 2020.
- Muller, K., & Schwarz, C., ‘Making America Hate Again,’ *SSRN Working Paper*, 2018.

- 
- Ngoc, H.L., Ngo, Q.H., Nuguyen, T.L., & Nguyen, Q.T., 'Fine-Tuning BERT for Sentiment Analysis of Vietnamese Reviews,' arXiv preprint, arXiv: 2011.10426, 2020.
- Norregaard, J., Adali, S., & Horne, B.D., 'NELA-GT-2018: a Large Multi-Labelled News Dataset for the Study of Misinformation in News Articles' *Proceedings of the International AAAI Conference on Web and Social Media*, Vol. 13, 2019.
- Official Journal of the European Union L 119/1 of 27 April 2016.
- Page, L.B., 'The PageRank Citation Ranking : Bringing Order to the Web,' *The Web Conference*, 1999.
- 'Quote Tweets,' *X Developer Platform*, n.d.
- Rasul, K., Rogge, N., & Simhayev, E., 'Yes, Transformers are Effective for Time Series Forecasting (+ Autoformer),' *Hugging Face*, 2023.
- 'Right-wing Extremism,' *Bundesamt für Verfassungsschutz*, n.d.
- 'Search Tweets,' *X Developer Platform*, n.d.
- Torregrossa, J., Bello-Orgaz, G., Camacho, D., Del Ser, J., & Martinez-Camara, E., 'A Survey on Extremism Analysis using Natural Language Processing: Definitions, Literature Review, Trends and Challenges,' *Journal of Ambient Intelligence and Humanized Computing*, Vol. 14, 2022, pp. 9869–9905.