UCLouvain

École polytechnique de Louvain

An LSTM approach to Predict Migration based on Google Trends

Authors: **Nicolas Golenvaux, Pablo Gonzalez Alvarez**
Supervisors: **Harold Silvère Kiossou, Pierre Schaus**
Readers: **Frédéric Docquier, Siegfried Nijssen**
Academic year 2019–2020
Master [120] in Computer Science and Engineering
Master [60] in Computer Science

An LSTM approach to Predict Migration based on Google Trends

Nicolas GOLENVAUX & Pablo GONZALEZ ALVAREZ

June 14, 2020
Version: Final Version

Université Catholique de Louvain

École Polytechnique de Louvain
Computer Science and Engineering Department
Master [120] in Computer Science and Engineering
Master [60] in Computer Science

Master Thesis

An LSTM approach to Predict Migration based on Google Trends

Nicolas Golenvaux & Pablo Gonzalez Alvarez

Supervisor
Harold Silvère Kiossou

Supervisor
Prof. Pierre Schaus

Reader
Prof. Frédéric Docquier

Reader
Prof. Siegfried Nijssen

June 14, 2020

Nicolas Golenvaux & Pablo Gonzalez Alvarez
An LSTM approach to Predict Migration based on Google Trends
Master Thesis, June 14, 2020
Supervisors: Harold Silvère Kiossou and Prof. Pierre Schaus
Readers: Prof. Frédéric Docquier and Prof. Siegfried Nijssen

Université Catholique de Louvain
Master [120] in Computer Science and Engineering
Master [60] in Computer Science
Computer Science and Engineering Department
École Polytechnique de Louvain
Rue Archimède 1 - Bte L6.11.01
B-1348 Louvain-la-Neuve

Abstract

Being able to model and predict international migration as precisely as possible is crucial for policy making. Recently Google Trends data in addition to other economic and demographic data have been shown to improve the prediction quality of a gravity linear model for the one-year ahead predictions. In this work, we replace the linear model with a long short-term memory (LSTM) approach and compare it with two existing approaches: the linear gravity model and an artificial neural network (ANN) model. Our LSTM approach combined with Google Trends data outperforms both these models on various metrics in the task of predicting the one-year ahead incoming international migration to 35 Organisation for Economic Co-operation and Development (OECD) countries: for example the root mean square error (RMSE) and the mean average error (MAE) have been divided by 5 and 4 on the test set. This positive result demonstrates that machine learning techniques constitute a serious alternative over traditional approaches for studying migration mechanisms.

Résumé

Il est essentiel pour l'élaboration des politiques de pouvoir modéliser et prévoir les migrations internationales aussi précisément que possible. Récemment, il a été démontré que les données de Google Trends, en plus d'autres données économiques et démographiques, améliorent la qualité des prévisions d'un modèle linéaire gravitationnel pour les prévisions à un an. Dans ce travail, nous remplaçons le modèle linéaire par une approche long short-term memory (LSTM) et nous le comparons à deux approches existantes : le modèle de gravité linéaire et un modèle de réseau neuronal artificiel (ANN). Notre approche LSTM, combinée aux données de Google Trends, surpasse ces deux modèles sur divers points pour la prévision à un an des migrations internationales entrantes dans 35 pays de l'Organisation de coopération et de développement économiques (OCDE) : par exemple, l'erreur quadratique moyenne (RMSE) et l'erreur moyenne (MAE) ont été divisées par 5 et 4 sur l'ensemble de test. Ce résultat positif démontre que les techniques d'apprentissage machine constituent une alternative sérieuse aux approches traditionnelles pour l'étude des mécanismes de migration.

Acknowledgement

Louvain-la-Neuve, June 14, 2020

During the journey that led to this thesis, many people have supported us. First and foremost, for their patience, trust, technical advises, and insightful discussions, we would like to sincerely thank both our supervisors Pierre Schaus and Harold Silvère Kiossou. We would not have written a preprint without you.

We are grateful and thankful to both Frédérique Docquier and Siegried Nijssen, for accepting to read this thesis as well as to be part of our jury during the thesis defence.

Next, we would like to thank André Gröger for providing the data from the "*Searching for a better life*" paper, which accelerated our work vastly.

Moreover, for is helpful inputs on both our experiments and the preprint, we would like to thank Vinasetan Ratheil Houndji.

For his seminars and excellent coaching session on how to "*Present your master thesis with impact*", a special thanks is given to Francesco Contino.

Finally, for their unconditional moral support, we would like to warmly thank our family and our friends.

Nicolas Golenvaux & Pablo Gonzalez Alvarez

Caminante, son tus huellas
el camino y nada más;
caminante, no hay camino,
se hace camino al andar.
Al andar se hace camino,
y al volver la vista atrás
se ve la senda que nunca
se ha de volver a pisar.
Caminante, no hay camino,
sino estelas en la mar.

Wanderer, your footsteps are
the road and nothing more;
wanderer, there is no road,
the road is made by walking.
Walking makes the road,
and turning to look behind
you see the path that you
will never tread again.
Wanderer, there is no road,
only foam trails on the sea.

— **Antonio Machado**

From Proverbios y cantares*, in Campos de Castilla, 1912,*
translated by Stanley Appelbaum*, Dover Publications, 2007.*

Machine Learning

Title text: *The pile gets soaked with data and starts to get mushy over time, so it's technically recurrent.*

—XKCD [75]

Contents

List of Figures

List of Tables

List of Algorithms

Introduction

Motivation and Problem Statement

Mobility has always been part of human history. In 2017, there were about 258 million international migrants worldwide, of which 150.3 million were migrant workers [70]. Modelling and forecasting human mobility is therefore important not only to help formulate effective governance strategies but also to deliver insight and scalable options to humanitarian responders and policymakers. However, developing reliable forecasting methods able to predict $T_{i,j}$, the number of people from a given region i to another region j is extremely challenging due to the absence of, low frequency and long lags in recent migration data, especially for developing countries [4, 9].

One way to mitigate this lack of timely data is the use of real-time geo-referenced data on the internet like the Global Database of Events, Language, and Tone (GDELT Project) or Google Trends. Both have been successfully used to make forecast in various fields [12, 1, 29]. Recently, Böhme et al. [9] demonstrated that adding geo-referenced online search data to predict migration flows yields better performance compared to only using common economic and demographic indices like gross domestic product (GDP), and population size. The authors propose to predict bilateral migration flows of the following year with a linear model relying on the Google trends data captured the previous year.

Results

In this master thesis we use exactly the same data, but we replace the linear model by a recurrent neural network (LSTM [37]) that is able to consider the whole history to make predictions. The focus of the prediction is on incoming international migration to 35 Organisation for Economic Co-operation and Development (OECD) countries. We demonstrate that the quality of the prediction can be drastically improved by capturing better complex migration dynamics [50] and complex interactions between the many features.

Thesis Structure

The outline of our work is the following.

Chapter 1: Context

Chapter 1 is a short attempt to give a picture of what is defined as migration, and gives some key numbers of both migration stocks and flows. It aims at providing the reader with an overview and some references to go into more detail about the complexity of studying migration.

Chapter 2: Related Works

Chapter 2 gives an overview of what has been previously done both using geo-referenced online data and in migration forecasting. The chapter starts by focusing on tools like Google Trends (GT) and the Global Database of Events, Language, and Tone (GDELT) project. We then present methods by which human mobility is traditionally predicted, using the gravity model and the more recent radiation model. Finally, we present what we believe to be a first attempt to use machine learning to predict human migrations.

Chapter 3: Background

Chapter 3 provides the background needed to understand the approach we present in the next chapter. We first introduce the Google Trends Index (GTI) feature. We give a few details on estimating the aforementioned gravity model using ordinary least squares (OLS). We then go deeper into neural networks and explain the different parts of an artificial neural network (ANN). We examine how a recurrent neural network can be built. Finally, we present the long short-term memory (LSTM), a gated recurrent neural network (RNN).

Chapter 4: Approaches

Chapter 4 describes in detail the approaches driving this work. It first presents the data set as well as previously used metrics for migration forecasting. It then provides more details about each approach: (a) the gravity model; (b) the ANN model; and (c) the LSTM model. For each model, we describe the specific architecture and how we treat the input features. Finally, specifically for the two deep learning techniques, we describe the training required and validation approach we adopted.

Chapter 5: Results and Discussion

Chapter 5 evaluates and compares our LSTM approach with the gravity and ANN approaches, two previously used methods in migration forecasting. It aims to show that by combining cutting edge deep learning techniques with geo-referenced online data like Google Trends, it is possible to build a model that results in better predictions by taking into account the complexity of the dynamics of human mobility.

1 Context

1.1 Migration Stocks and Flows

The International Organisation for Migration (IOM) [40] gives the following definitions for migration and migrant:

Migration "*The movement of persons away from their place of usual residence, either across an international border or within a State*" [40, p. 135];

Migrant "*An umbrella term, not defined under international law, reflecting the common lay understanding of a person who moves away from his or her place of usual residence, whether within a country or across an international border, temporarily or permanently, and for a variety of reasons. The term includes a number of well-defined legal categories of people, such as migrant workers; persons whose particular types of movements are legally defined, such as smuggled migrants; as well as those whose status or means of movement are not specifically defined under international law, such as international students*" [40, p. 132].

Although the first definition is not universally agreed on, it is widely accepted in different settings and is the one recommended for statistics by the United Nations Department of Economic and Social Affairs (UN DESA) [69]. From this definition, migration can be of two kinds: (a) international migration; or (b) internal migration. In this thesis, we focus on inflows of legal international migrants towards OECD countries [54].

According to the Internal Displacement Monitoring Centre (IDMC) [38], there were about a total of 50.8 million internal displacements in 2019. Of these, 45.7 million are as a result of conflict and violence, while 5.1 million were caused by natural disasters. If we focus on the flows in 2019, there were 33.4 million new displacements: 8.1 million driven by increasing levels of violence in Burkina Faso, Yemen, and Libya; while 24.9 million new displacements were mainly due to geophysical or weather-related disasters.

What about international migration? According to the IOM [41], there were about 279 million international migrants worldwide in 2019. Notice that this was about

3.5% of the 2019 world population: the overwhelming majority of people stay in their country of birth. More than half of all international migrants (141 million) lived in Europe or North America. If we focus on the flows, there were about 7.06 million incoming international migrants to the 35 OECD countries in 2016. Notice that the most recent numbers are from 2016 and do not cover all the countries. The two most reliable sources are: (a) UN DESA's International Migration Flows dataset [71]; and (b) OECD's International Migration Database [54]. However, migration flow data is hard to come by since official data are published at low frequency and with long lags. As the IOM (2020) states: "*migration is notoriously difficult to predict with precision because it is closely connected to acute events (such as severe instability, economic crisis or conflict) as well as long-term trends (such as demographic change, economic development, communications technology advances and transportation access)*"[41, p. 3].

Although the data set used in this thesis focuses on international migration flows from countries towards OECD countries, it is important to underline that most international migration takes places within particular regions of the world. It has been clearly shown that most migratory movements are intra-regional as is the case within Africa, Asia, and Europe. However, this is not the case for Latin America, the Caribbean and Northern America, where most migrants reside outside their birth regions. This demonstrates a complex dynamic in human mobility and debunks a myth: most international migration is in fact not directed towards Europe or North America [41, 68].

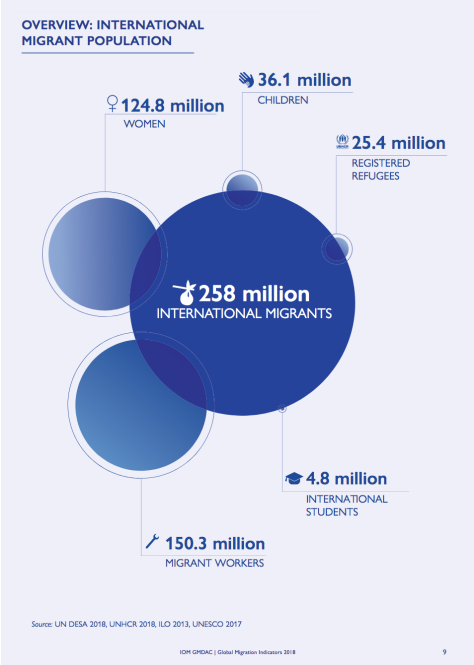

Fig. 1.1.: International Migrant Population in 2017 [39].

Why do people move? The main reason is economic due to lack of employment or other professional opportunities, with most of those who move being migrant workers as can be seen in Figure 1.1. Other reasons may include: family reunion and joining diaspora communities, climate change (deforestation, sea level rise, etc.), war, and epidemics [41].

2 Related Works

2.1 Use of Geo-referenced Online Search Data in Forecasting

In 2019, there were about 4.39 billion internet users and 3.48 billion people using social networks which make respectively 67% and 45% of the worldwide population and these numbers tend to increase by 10% each year [44]. In January 2019 a survey [6] showed that a person spent an average of 3 hours and 28 minutes per day browsing the internet and *Google Search Statistics - Internet Live Stats* [33] records more than 3.5 billion searches per day.

With the increased usage of online search engines, social networks, online shopping and Internet in general, geo-referenced online search data have become more and more appealing and widespread for all kind of businesses and research in economics and more broadly in social sciences [19]. Indeed such data portrays an increasingly important part of our everyday life and represent an inexhaustible source of real-time information that can be used to measure, compare, nowcast (i.e., "*obtaining relevant information much earlier than through traditional data collection techniques*" [4, p.1]), and even forecast a very wide range of human behaviours around the world [72].

The ease of access to this kind of resources combined with technical and technological advancements as in machine learning, which allows to analyse more and more data, has allowed a growing literature using geo-referenced online data to predict social or economic outcomes in very various fields. Choi and Varian [12] showed how Google Trends tend to enhance nowcasting on economic indicators like car sales, travel destination planning or unemployment claims.

One of the most prominent applications so far has been published by Ginsberg et al. [29], presenting a method of analysing large numbers of Google search queries to track influenza-like illness in a population. They predicted weekly levels of influenza activity based on the Google Flu Trend indicators with a reporting lag of only about one day.

Askitas and Zimmermann [4] highlights the use of internet data for social sciences and more particularly on human resources issues by presenting a selection of relevant literature and discussing the challenges of online data for these topics. One of the most encountered applications for the moment is the use of Google Trends data to predict unemployment which has been the subject of study in Germany [5], in France [22] and in the United States of America [14].

The use of internet metadata in the field of human migration has only been recently explored by a small number of applications. This kind of real-time geo-referenced data is even more useful in the area of nowcasting and forecasting migration flows because it can be used to mitigate the lack of or the lag in data from which migration data suffers strongly [9].

Ahmed et al. [1] presented a system combining several information sources, including the GDELT (Global Database of Events, Language, and Tone) project, for mass-migration forecasting using the 2015 European refugee crisis as a case study. More recently, Böhme et al. [9] demonstrated that adding geo-referenced online search data to traditional models predicting migration flows yields better performance compared to only using common economic and demographic indices, for example, gross domestic product (GDP) and population size.

2.2 Traditional Models

To model human mobility, the aim is to predict $T_{i,j}$, that is, to find an estimation $\hat{T}_{i,j}$, the number of people moving from a given region i to another region j among m origin regions and n destination regions. Traditional and ML techniques have each a different approach to the problem. In traditional models, the problem is divided into two subproblems: (a) predict G_i the number of people leaving a region i, also known as the production function; and (b) predict $P_{i,j}$ the probability of a movement from i to j. Thus we have $\hat{T}_{i,j} = G_i P_{i,j}$.

2.2.1 Gravity Model

The first gravity models for human migration can be dated back to 1885 and were known as the laws of migration [58, 57], as pointed out by Anderson [2] in his review of the gravitational model. Gravity models, inspired by Newton's law, is where the probability of a movement between two regions is both proportional to

the population of the two regions i and j, and inversely proportional to the distance between them [47, 56]. The most used version of the gravity model is:

$$T_{i,j} = C\frac{m_i^{\alpha} \times n_j^{\beta}}{r_{i,j}^{\gamma}} \tag{2.1}$$

$T_{i,j}$ is the number of people who moved from a region i to a region j. m_i, respectively n_j, is the size of the population of region i, respectively j. $r_{i,j}$ is the distance between the two regions. C is a normalisation constant. α, β, and γ are the parameters to be estimated. Taking the base 10 logarithm on each side yields:

$$\log\left(T_{i,j}\right) = \log\left(C\right) + \alpha\log\left(m_i\right) + \beta\log\left(n_j\right) - \gamma\log\left(r_{i,j}\right) \tag{2.2}$$

Thus, the parameters can be estimated through a linear least square methods, like the scaled ordinary least squares (OLS), although other ways can be used too [21]. Chapter 3 will explain how to estimate the gravity model using the OLS method.

Albeit their large use, gravity models have a tendency to not model properly long distance mobility [50]. Radiation models are a proposed response.

2.2.2 Radiation Model

Radiation models, inspired by diffusion dynamics, is where a movement is emitted from a region i and has a certain probability of being absorbed by a neighbouring region j [65]. The most used version of the radiation model follows:

$$T_{i,j} = T_i\frac{m_i n_j}{(m_i + s_{i,j})(m_i + n_j + s_{i,j})} \tag{2.3}$$

Like the gravity model, $T_{i,j}$ is the number of people who moved from a region i to a region j. It is estimated as a fraction of T_i the travellers from population i. The subtlety here is that this probability is dependent on the population of origin m_i, the population of destination n_i, and on the population $s_{i,j}$ inside a circle centred in i with a radius $r_{i,j}$ equal to the distance from i to j[1]. Taking the base 10 logarithm on each side yields:

$$\begin{aligned}\log\left(T_{i,j}\right) = &\log\left(T_i\right) + \log\left(m_i\right) + \log\left(n_j\right) - \log\left(m_i + s_{i,j}\right) - \\ &\log\left(m_i + n_j + s_{i,j}\right)\end{aligned} \tag{2.4}$$

[1]Origin and destination populations are not included.

In his review of both models, Masucci et al. [50] shows that gravity has a better overall performance and tends to be better for short distance mobility, while radiation tends to be better for long distance mobility.

2.3 Machine Learning Models

With ML models, the approach is quite different as the goal is to directly predict $\hat{T}_{i,j} = f(features)$ from a set of features[2]. To the best of our knowledge, Robinson and Dilkina [60] is the fist attempt to use ML in order to predict human migration. The authors use two ML techniques: (a) "*extreme*" *gradient boosting regression (XGBoost)* model; and (b) deep learning based *artificial neural network (ANN)* model. The authors compared these techniques with both the gravity model and the radiation model, and find that these ML techniques outperform them on several commonly used metrics.

Similarly to us, this approach also attempts to directly predict $\hat{T}_{i,j}$ from the set of features without requiring any production function. But it exhibits two important differences with our method: a) it uses traditional features for their prediction model, which is composed of geographical and econometric properties such as the inter-country distance, median household income, etc.; and b) it does not capture the dynamic aspect since the prediction only relies on the previous time-step set of features.

2.3.1 Scalable Tree Boosting System

The XGBoost model is a "*scalable tree boosting system*" [11, p. 1]. It is based on gradient boosting [23] and is extensively used by winners of Kaggle competitions [10, 8]. One interesting consequence of using XGBoost is that it classifies the features according to their importance [11, 60].

2.3.2 Artificial Neural Network

The ANN model presented by Robinson and Dilkina [60] uses rectified linear unit (ReLU) activation layers. Its main particularity is its loss function, which is based on

[2]Notice that you could approach the problem the same way as with traditional models but it is not a common practice.

the *common part of commuters (CPC)* metric defined in detail in Chapter 4. The CPC is simply the percentage of correctly predicted numbers of travellers [46]. Chapter 3 will give more details on activation layers, loss functions and how an ANN works.

Background

3

3.1 The Google Trends Index

As Google is by far the most widely used search engine with more than 70% of the market share and more than 1 billion users worldwide[1], relying on its data seems like the most appropriate and general choice to represent the behaviour of the internet users around the world and to measure their migration intentions.

Böhme et al. [9] suggest a new measure of migration intentions using aggregate online search intensities: the *Google Trends Index (GTI)*. The GTI is based on the Google Trends (GT) data freely accessible at [32]. The GT tool allows collecting a daily measure of the relative quantities of web search of a precise keyword in a particular region of the world for a specified span of time[2] [25, 32]. In this thesis we make use of this new feature.

Figure 3.1 shows the result obtained on the GT website for the keyword *migration* in Belgium over a 12 months period.

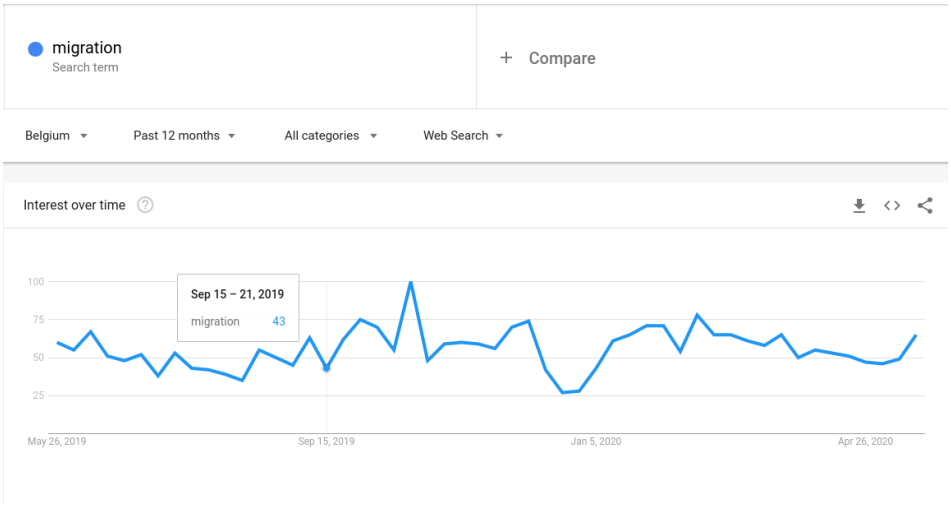

Fig. 3.1.: Result displayed by Google Trends for the keyword *migration* in Belgium from May 26, 2019 to May 25, 2020 [32].

[1] According to Net MarketShare: `https://netmarketshare.com/`

[2] The data can be downloaded from their website, or through an unofficial API [25].

GT provides two samples: (a) real-time data, "*covering the last seven days*"; and (b) non-real-time data, which "*goes as far back as 2004 and up to 36 hours before your search*" such as in Figure 3.1[3] [32]. The GTI makes use of the latter. GT provides search results that are "*are normalised to the time and location of a query by the following process*" [31, 61]:

- "*each data point is divided by the total searches of the geography and time range it represents to compare relative popularity. Otherwise, places with the most search volume would always be ranked highest*" [31];
- "*the resulting numbers are then scaled on a range of 0 to 100 based on a topic's proportion to all searches on all topics*" [31];
- "*different regions that show the same search interest for a term do not always have the same total search volumes*" [31].

Depending on the size of the period GT will provide a weekly index or a monthly index when the period is over 5 years long. For each index it will give a relative search result which oscillates between 0 and 100 using the aforementioned normalisation. It is important to notice that if the volume of searches for a certain keyword is too low, that is below an unknown threshold fixed by Google, the value of the returned index will be 0.

To best represent the migration intentions of Internet users via online searches, a set of terms related to the theme of migration is selected. It is composed of the 67 most semantically related terms to "*immigration*" and the 67 most semantically related terms to the word "*economics*" according to the website Semantic Link[4]. This website is based on the English version of the Wikipedia online encyclopedia to determine which words are semantically related. In addition to the 67 most semantically related terms to *immigration*, the set of keywords is also composed of the 67 most semantically related terms to *economics* because economy is the main factor that drives people to take migration decisions. Indeed, today the main reasons for migration are employment opportunities and higher wages [52, 62].

Every term is transcribed in 3 languages with Latin roots: English, Spanish and, French. This ensures a simpler data extraction while covering a maximum of people, that is, about 841 million native speakers [18]. Table A.1 in the appendix contain the set of main keywords for these 3 languages. In order to capture the whole keyword's predictive power, different versions of each word are included. The American and British English spellings of words are taken into account together with the singular

[3] In this work, when we mention Google Trends we refer to their non-real-time sample.
[4] `https://semantic-link.com/`

and plural forms. For the French and Spanish languages, the male and female forms are both considered as well as a version with accent and another without. The joint search intensities of the different forms are then captured by using the + operator between each version of the term which corresponds to a boolean OR operator in the Google Trends tool.

As Böhme et al. [9] use the data on human migration furnished by the OECD [54], which provides a yearly incoming migratory flow from 101 countries of origin to the 35 countries member of the OECD from the early 1980's until 2015, the Google Trends Indexes (GTI) must match these values.

The Google Trends Indexes of a precise keyword for a particular country are then calculated by capturing the measures provided by Google Trends for the chosen keyword in the geographical area corresponding to the country in question for the time period spanning from 2004 to 2014[5]. Since the values provided by Google are provided as intervals of one month[6] and are normalised in a range between 0 and 100, the GTI are computed by taking the average of the values for each year in order to match the migration data. The final indexes therefore reflect the variation of the quantity of searches for the keyword over the years in the specific country.

The bilateral GTI data is made up of the two different forms of vectors: $GTIbi_{i,j,t}$ and $GTIuni_{i,t} \times GTIdest_{i,j,t}$. Three different forms of GTI values are thus defined:

the vector of unilateral GTI ($GTIuni_{i,t}$) contains the GTI values of the set of keywords for the country of origin i during the year t (e.g., *visa*, *migrant*, *work*, etc.);

the vector of bilateral GTI ($GTIbi_{i,j,t}$) contains GTI values also specific to the country of destination j. The values are still captured in the country of origin i during the year t but the related keywords correspond to the combination of the terms from the migration-economics set with the name of the destination country (e.g., *visa Spain*, *migrant Spain*, *work Spain*, etc.);

the destination GTI ($GTIdest_{i,j,t}$) contains only the GTI value of the keyword corresponding to the destination country's name j (e.g., *Spain*) in the country i and the year t.

Appendix C on page 63 contains a description of our script extracting the different forms of GTI values. It is used in conjunction with the keywords in Appendix A on page 57. The script is available on the following repository:

[5] Google Trends data only starts from 2004 and the migration data stops after 2015.
[6] This is specific to requests spanning from 2004 to the present, for shorter periods of time (below 5 years), the intervals between each index are more frequent.

https://github.com/aia-uclouvain/gti-mig-paper

3.2 Estimating Gravity Models using Ordinary Least Squares

Taking the logarithm of the gravity model (refer to equation (2.2) on page 9) makes it possible to build an estimator, or a learning model, using ordinary least squares (OLS), the most commonly used linear least squares method [73].

Let Y be a response variable, and X a covariate, also known as predictor or a feature. The simple linear regression model is thus:

$$Y_i = \beta_0 + \beta_1 X_i + \varepsilon_i \qquad \forall i = 1, \ldots, n \tag{3.1}$$

Generalising to n observations with p parameters, we get the following matrix form:

$$\mathbf{Y} = \mathbf{X}^T \boldsymbol{\beta} + \boldsymbol{\varepsilon} \tag{3.2}$$

Where $\mathbf{Y}$ a n x 1 column vector corresponding to the response variable; $\mathbf{X}$ is a n x p matrix corresponding to the input features; $\boldsymbol{\beta}$ is a p x 1 column vector of the unknown parameters; $\boldsymbol{\varepsilon}$ a n x 1 column vector corresponding to the robust error term.

The idea of the ordinary least squares is to determine the weights $\boldsymbol{\beta}$ that minimise the residual sum of squares, as given by the following equation [73]:

$$\hat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} ||\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}||^2 \tag{3.3}$$

To introduce our approach, $\mathbf{Y}$ would be the next year migration predictions $\hat{T}_{i,j,t+1}$, while $\mathbf{X}$ are the input features from the current year of our different models (see Chapter 4 for more on that).

3.2.1 Estimating the Unilateral and Bilateral Gravity Models

Let us now look at a practical use of the gravity model with Google Trends data. Böhme et al. [9] demonstrate how the use of geo-referenced online search data

to measure the migration intentions brings strong additional predictive power for international migration flows compared to traditional models from the migration literature. Since their work is very recent and very close to the purpose of this thesis, we attached great attention to it. We already presented a part of the paper in the section 3.1 by describing the method employed to select the keywords linked to the migration topic as well as the method employed to extract the relative quantities of web searches via Google Trends. We already defined the different forms of the Google Trends Indexes used, thus in this section we will focus on the presentation of the different models together with their results.

The authors rely on two gravity models which are estimated by a linear regression, and more specifically using the scaled *ordinary least squares (OLS)* estimation.

Unilateral Model

Two major different migration models have been presented. The first one is unilateral: it tries to predict the outgoing flow of migrant from an origin country during a certain year, regardless of their destination, based on economic, demographic and geo-referenced online search data of the previous year.

The equation for the unilateral model is the following:

$$Y_{i,t+1} = \beta_1 \, GTIuni_{i,t} + \beta_2 \, O_{i,t} + \gamma_i + \delta_t + \varepsilon_{i,t} \tag{3.4}$$

From the above equation, the dependent variable $Y_{i,t+1}$ is the base 10 logarithm annual flow of migrants leaving the country of origin i towards any of the OECD's destination countries during the year $t+1$: $T_{i,j,t+1}$[7]. As said earlier, this data comes from the OECD [54] which provides a yearly incoming migratory flow from 101 countries of origin to the 35 countries member of the OECD from the early 1980's until 2015.

The vector of unilateral GTI or $GTIuni_{i,t}$ contains the GTI values of the 134 keywords of the "*immigration-economic*" set described earlier for the country of origin i during the year t (as explain in section 3.1).

Thereafter, $O_{i,t}$ is a vector of control variables of the country of origin and destination for the year t. These variables contain different demographic and economic indicators about the country for the concerned year inspired by the existing literature

[7] To avoid computation issues with null migration flows, the logarithm is calculated on the value plus one: $Y_{i,j,t+1} = log(T_{i,j,t+1} + 1)$. This applies to any other variable resulting from a logarithmic calculation.

on migration intentions. They gather as basic measures the base 10 logarithm of the total GDP and the base 10 logarithm of the population. However, they also include additional measures like the unemployment rate, the share of the young population, the share of the internet users (per 100 people), the share of mobile phone subscriptions (per 100 people). These first control variables are collected from the World Development Indicators [74]. Other factors from various sources are also appended like the number of weather and non-weather disasters [20], the Polity IV Autocracy Score and the State Fragility Index [49]. Finally, since the used keywords are translated in only 3 languages (English, Spanish, and French), the shares of the native population that fluently speaks these languages are also included [51]. These control variables are added in extension because many of those data are missing for some origin countries.

Afterwards γ_i and δ_t are vectors of fixed-effects specific respectively to the origin country i and to the year t. The first vector absorbs time-invariant factors specific to the country of origin while the second vector absorbs time-varying factors like economic crisis, population dynamics, etc. Finally $\varepsilon_{i,t}$ represents the robust error term.

Bilateral Model

The second model and the most interesting one is the bilateral model: it no longer considers only the outgoing migration flow from a country of origin but rather the number of migrant leaving a country of origin towards a specific country of destination during a certain year. The equation for the bilateral model is the following:

$$\begin{aligned} Y_{i,j,t+1} = \beta_1\, GTIbi_{i,j,t} + \beta_2 GTIuni_{i,t} \times GTIdest_{i,j,t} + \\ \beta_3\, O_{i,t} + \beta_4\, D_{j,t} + \gamma_{i,t} + \delta_{j,t} + \tau_{i,j} + \varepsilon_{i,j,t} \end{aligned} \tag{3.5}$$

The dependent variable $Y_{i,j,t+1}$ is the base 10 logarithm annual flow of migrant from the country of origin i to the country of destination j during the year $t+1$ (plus one). These values come from the same OECD migration database as for the unilateral model.

A crucial difference with the precedent model is that this one includes different forms of the GTI vectors. Indeed, since bilateral GTI $GTIbi$ uses more complex keywords (2 terms instead of 1) 94% of the total corresponding GTI scores are null because the search volume is not important enough to pass the Google Trends threshold. The $GTIuni_{i,t} \times GTIdest_{i,j,t}$ parameters provide thus a less relevant but

more flexible and less sparse form of the bilateral GTI values (34% of null GTI scores on the whole database).

$O_{i,t}$ and $D_{j,t}$ are the vectors of the control variables specific to the country of origin and to the destination country for the year t. They contain the same basic and extended parameters as in the unilateral model.

$\gamma_{i,t}$, $\delta_{j,t}$ and $\tau_{i,j}$ are vectors of fixed-effects specific respectively to the pair origin-year, to the pair destination-year and to the pair origin-destination. They absorb time-varying factors like economic crisis, population dynamics, policy changes but also time-invariant factors such as distance between the two countries, their common languages, etc. And again the $\varepsilon_{i,j,t}$ represents the robust error term.

Results

Multiple experiments comparing the quality of predictions of different versions of the unilateral and bilateral model are then carried out. Among those different variations applied to the initial models, we can cite a version with the totality of the control variables (the basic model only uses the GDP and the population size), regressions that are only fitted with a part of the total observations, and models with different types of fixed-effects vectors. For each variant, a model not using the GTI vectors is confronted with another regression using those vectors.

It is important to notice that every model is trained via a scaled *ordinary least squares (OLS)* and that no validation or test sets are defined, the models are evaluated with the same data as those used for their training. Moreover, the metric used to evaluate the quality of the predictions is the coefficient of variation R^2. A second metric called the within-R^2 is also calculated, it corresponds to the coefficient of variation from the mean-deviated regression, that is, the ordinary R^2 from running OLS on the transformed data.

The results of these different experiments show that the within-R^2 is always significantly increasing when we add to the models the GTI vectors whatever the version of the initial model. As for the overall R^2 it increases slightly or remains constant when adding the GTI. The results of these experiments are presented in Figures B.1 and B.2 in the Appendix.

To conclude, Böhme et al. [9] presented evidence that adding information about internet search volumes for specific keywords can help estimate international migration flows using traditional models. Furthermore, they found that these improvements are even more significant when we limit the predictions to countries of origin with a

high internet penetration and whose population mainly speaks the languages used for the extraction of the online search intensities. Finally they also showed that the used GTI reflects genuine migration intentions by comparing the predictive power of migration survey data with the one using GTI.

3.3 Artificial Neural Networks and Deep Learning

We will focus on Artificial Neural Networks (ANN) and specifically on Recurrent Neural Networks (RNN) using Long-Short Term Memory (LSTM) cells, since it seems to be a promising technique. To better understand the LSTM, let us proceed in three steps: (a) understand what an ANN is; (b) explain how RNN came by and how they work; and (c) what problems the LSTM addresses and how it works. Notice that throughout this section it is important to remember Occam's razor principle: among all theories explaining the world equally well, the simplest is the best.

3.3.1 Artificial Neural Networks

The Artificial Neural Network (ANN) is an information processing paradigm that is inspired by the way the biological nervous system such as brain process information. It is composed of large number of highly interconnected processing elements (neurons) organised in layers working in unison to solve complex problems [48]. It is able to catch the complex dynamics behind the migrations [60].

An ANN goal is to compute an estimation function, for instance the next-year migration flows $\hat{T}_{i,j,t+1}$, from a certain input $\mathbf{x}$. In its most basic form it has three layers: (a) an input layer; (b) a hidden layer; and (c) an output layer. One special type is the feedforward ANN, where information flows from the input layer, through the hidden layer and finally to the output layer. This process is called forward propagation. It has thus no feedback connections. Historically the feedforward ANN is known as the single layer perceptron.

Information Flow and Activation Layer

A feedforward ANN is formed by a network of functions, or layers, that is $\hat{T}_{i,j} = T^{(3)}_{i,j}(T^{(2)}_{i,j}(T^{(1)}_{i,j}(\mathbf{x})))$. $T^{(1)}_{i,j}$ is the first layer of the network, $T^{(2)}_{i,j}$ is the second layer, and $T^{(3)}_{i,j}$ is the third layer. This can be visualised in Figure 3.2. Together they form a

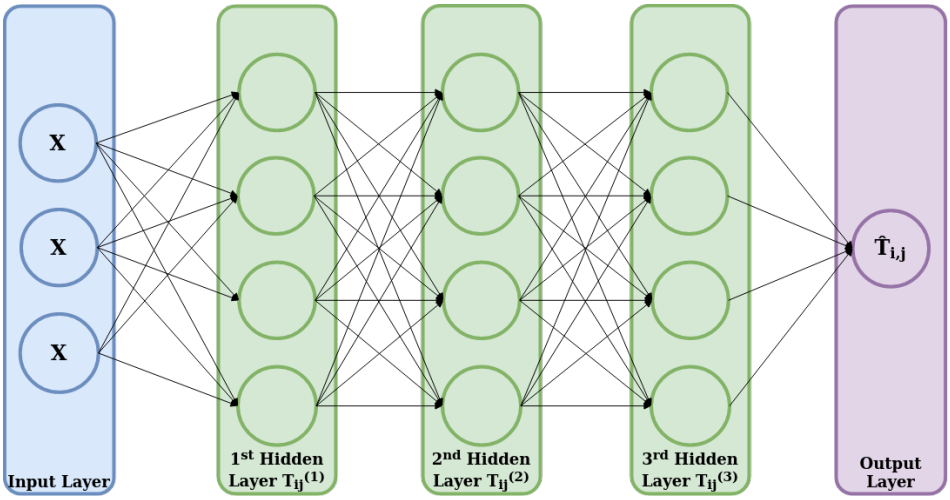

Fig. 3.2.: A feedforward ANN of width 4 and depth 3 with 3 inputs and 1 output.

chain of functions known as the *network*. The length of this chain is known as the *depth* of the model, while the dimensionality of each hidden layer is called the *width* of the model. These layers can be also thought of as units. To compute the hidden layer values, you need to choose the *activation function*.

Three common activation functions are the *hyperbolic tangent*, the *logistic*, and the *rectifier* as can be seen in Figure 3.3. Each one has its advantages and drawbacks [30].

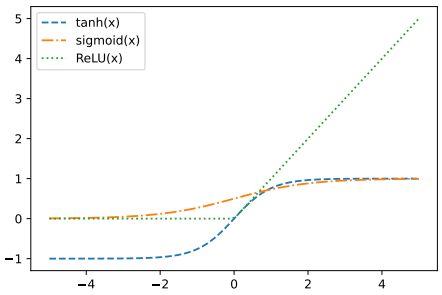

Fig. 3.3.: The hyperbolic tangent, the logistic (or sigmoid), and the rectifier (or ReLU) activation functions.

The hyperbolic tangent has a range between -1 and 1. It has the interesting property to saturate for arbitrarily large positive or negative values. Thus it is insensitive to small input changes, but is particularly useful to detect a sign change. It may give rise to the vanishing gradient problem. It has the following definition and derivative:

$$t(x) = tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{3.6}$$

$$t'(x) = 1 - t(x)^2 \tag{3.7}$$

The logistic better known as the sigmoid, has a range between 0 and 1. It has mostly the same properties than the hyperbolic tangent but it is not centred in

0. It may also give rise to the vanishing gradient problem. It has the following definition and derivative:

$$t(x) = \sigma(x) = \frac{1}{1 + e^{-x}} \tag{3.8}$$

$$t'(x) = t(x)(1 - t(x)) \tag{3.9}$$

The rectifier also known as the rectified linear unit (ReLU), is the most recommended activation function for most feedforward artificial neural networks. Although this function yields a nonlinear transformation, it keeps many nice properties of linear models, including for generalisation, since it is piecewise linear. Thus it is easily optimised using gradient-based approaches. However, it may give rise to the exploding gradient problem. It has the following definition and derivative:

$$t(x) = \begin{cases} 0 \text{ for } x \leq 0 \\ x \text{ for } x > 0 \end{cases} \tag{3.10}$$

$$t'(x) = \begin{cases} 0 \text{ for } x \leq 0 \\ 1 \text{ for } x > 0 \end{cases} \tag{3.11}$$

Backpropagation, Optimisation and Loss Functions

As a machine learning technique, a neural network is trained using gradient descent. Backpropagation is the process through which the gradient is computed by sending the flow of information backwards through the network. A specificity of neural networks is its tendency to make most loss functions become nonconvex. Thus iterative and optimised gradient descent are used. The most common one is the Stochastic Gradient Descent (SGD) [63]. The Adaptive Moment Estimation (Adam) is a variation for SGD where the learning rate is adapted using both the gradient and the second moments of the gradient [45].

To learn properly from their errors, neural networks also need a loss function. A loss function $f(T, \hat{T})$ is a method evaluating how well the algorithm models the data, it calculates the error between the predicted output value $\hat{T}$ and the ground truth value T. This means that the lower the result of the function, the more the model fits the data. Both the ANN and the LSTM used in this paper uses Robinson and Dilkina [60] custom loss function as well as two more conventional loss functions. Let $\mathbf{T}$ be the ground truth value, $\hat{\mathbf{T}}$ the prediction matrix, m the number of origin countries, n the number of destination countries, $v_j = \sum_{i=1}^{m} T_{i,j}$ the number of

incoming migrants for a zone j, $\hat{v}_j$ its prediction. The different loss functions are thus given by:

Common Part of Commuters This is is based on the Common Part of Commuters metrics proposed by Robinson and Dilkina [60]. It simply compares the ground truth value with the predicted migration values. For more details refer to the explanation of equation (4.1) on page 31.

$$L_{CPC}(\mathbf{T}, \hat{\mathbf{T}}) = 1 - \frac{2\sum_{i,j=1}^{m,n} \min\left(T_{i,j}, \hat{T}_{i,j}\right)}{\sum_{i,j=1}^{m,n} T_{i,j} + \sum_{i,j=1}^{m,n} \hat{T}_{i,j}} \tag{3.12}$$

Mean Absolute Error Which is the sum of the absolute differences between the ground truth and predicted migration values.

$$L_{MAE}(\mathbf{T}, \hat{\mathbf{T}}) = \frac{1}{m \cdot n} \sum_{i,j=1}^{m,n} |T_{i,j} - \hat{T}_{i,j}| \tag{3.13}$$

Mean Square Error Which is the average squared differences between the ground truth and predicted migration values.

$$L_{MSE}(\mathbf{T}, \hat{\mathbf{T}}) = \frac{1}{m \cdot n} \sum_{i,j=1}^{m,n} (T_{i,j} - \hat{T}_{i,j})^2 \tag{3.14}$$

Generalisation and Regularisation for Deep Learning

To build a good model, both the *training error*, made during the training, and the *generalisation error*, made during the testing, need to be minimised. When training the model it is important to avoid *underfitting*. Underfitting means a model is not complex enough to adequately capture the underlying relations of the data, that is, the training error is large. Moreover, when testing the model, it is also important to avoid *overfitting*. Overfitting means that the model performs very well on the training data since its production fit very closely to this particular data but it will give poor results on new data, that is, the model will not generalise well on previously unseen or future data. It can be caused by an overly complex model [30].

Compared to other machine learning techniques, deep learning has a stronger tendency for overfitting. It is thus crucial to use methods to identify and avoid it. One way to regulate the complexity of the model is the use of regularisation techniques. One of the most successful used strategy is the dropout where a fixed proportion p of input features or of hidden units are randomly dropped during the training. Dropout

is an elegant technique, due both to its simplicity and computational efficiency, which leads to better performances. Nonetheless, when only a few labelled examples are available it can be less effective[66].

Limitations

Although its many applications, an ANN has two constraints [34, 30]: (a) it only takes a fixed set of vectors as input and returns a fixed set vectors as output; and (b) it does not share features learned across different parts of the sequential data, that is, it cannot learn from previous examples to inform later ones. One way to solve that is to extend the ANN with loop connections: the recurrent neural network (RNN).

3.3.2 Recurrent Neural Networks

A *recurrent neural network (RNN)* is more flexible than an ANN since [34, 30]: (a) it allows to operate over sequence of data where the inputs and the outputs can be of varying length; and (b) it allows information to persist through internal loops. Thus to put it simply, a RNN is an ANN with recurrent connections that make it possible to memorise previous inputs in the network's internal state, which in turn can be used to influence the network's output.

This can be visualised in Figure 3.4. It is a RNN with input $X_{i,j,t}$, with a state corresponding to the hidden layers, and an output $\hat{T}_{i,j,t}$. It shows how the information is passed forward through time. The left side can be unfolded to give the right side where each node is associated to a specific time step.

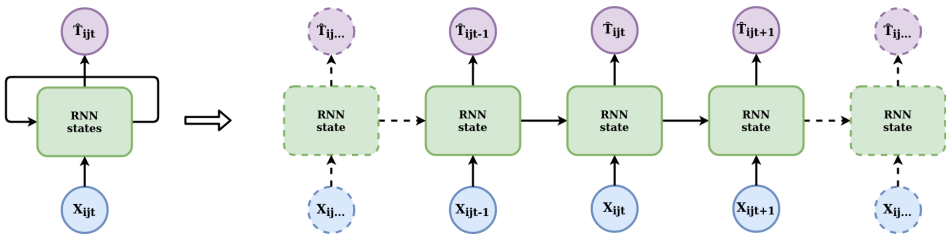

Fig. 3.4.: An unfolded RNN. Each state is a neural network that can be trained. The left-side correspond to the rolled RNN, while the right-side to the unrolled RNN.

Limitations

Albeit its ability to learn from the past, a RNN is not capable to learn properly long-term dependencies due to both the vanishing gradient problem and the exploding gradient problem [34, 30]. This is were the LSTM comes in.

3.3.3 Long Short-Term Memory

A set of various approaches, known as gated RNN, have been proposed to make RNN handle long-term dependencies. Two of the most known approaches are the Long Short-Term Memory (LSTM) and the Gated Recurrent Unit (GRU), which are, to this day, the most effective sequence models used in practical applications [30]. Since its first publication, LSTM has gained momentum for several applications, including in forecasting, and has been shown to yield significantly better performances in the prediction of time series compared to other ML techniques [64, 35, 26, 76, 67, 48, 28, 42]

Historically the original, or standard, LSTM, was a response to RNN's inability to learn long-term dependencies, due to both the exploding gradient problem and the vanishing gradient problems [36, 17, 7, 34]. This standard LSTM introduced two gates into the memory cell, the input and output gates, as well as an internal loop, known as the constant error carousel (CEC). The input gate, respectively the output gate, purpose is to determine which signals should enter, respectively leave, while the CEC's is to keep it in memory. Together they control the constant error flow of a cell [37].

However the CEC may actually break the network, that is, the conjunction of various standard LSTM memory cells, since the internal state of an individual memory cell may grow indefinitely. To solve that, the forget gate was introduced, which makes the CEC dynamic by giving it weights. This dynamic CEC is thus able to learn to reset its weight at the appropriate times and the gate is thus also known as the reset gate. Moreover, it is advocated to add a bias of 1 to the LSTM forget gate. In deed, recently it has been even strongly advised since it has been shown to significantly increase performances on tasks where other gated-RNN techniques were ahead of it [26, 42].

Thus we have the basic building block of our LSTM: "*one or more self-connected memory cells and three multiplicative units—the input, output and forget gates—that provide continuous analogues of write, read and reset operations for the cells*" [34, p.31]. Each of these gates are a neural network that can be trained. The input

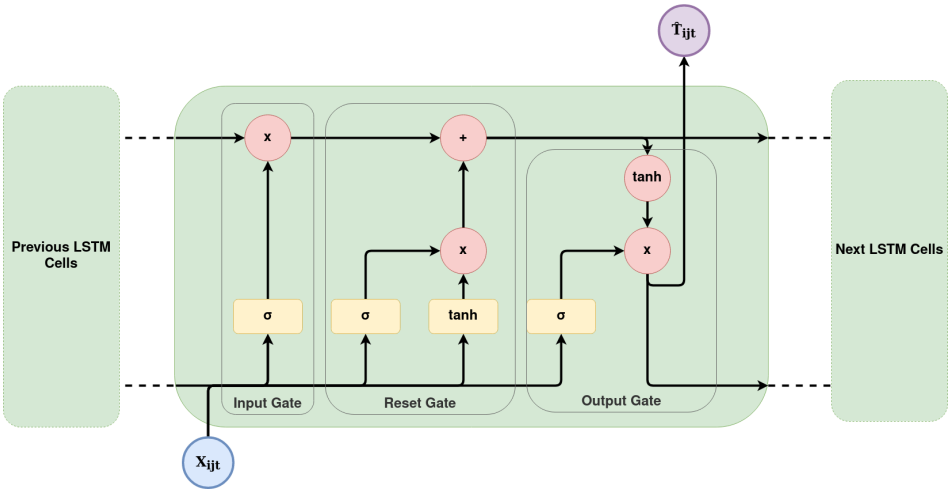

Fig. 3.5.: LSTM gated-RNN network with a focus on 1 cell. Each cell has 3 interacting gates, the yellow boxes from left to right: the input gate (σ), the forget gate (σ and $\tanh$), and the output gate (σ). Each gate is a neural network that can trained. Inspired by Olah [55].

gate learns what information to keep in memory, the forget gate how long it should be kept, and the output gate when to output it. This upgraded standard LSTM is known as the extended LSTM [26]. Figure 3.5 shows the basic of a LSTM recurrent network.

Finally, another improvement is the use of the dropout, a regularisation technique that randomly drops a fraction of input features or hidden units. It has been shown to improve drastically performances of neural networks, even for gated RNN like GRU or LSTM [66, 24].

Some other variants exists, and one of the most promising one introduces peepholes inside of the memory cell [27]. We favour the aforementioned upgraded version, that is the extended LSTM with dropout regularisation, since it is simpler while keeping its competitors at bay[8] [42, 35, 30]. Moreover, it is also the one implemented by the Keras neural network library, which we use in this work [13].

[8]Remember Occam's razor principle.

4 Approaches

In order to demonstrate that machine learning techniques along with geo-referenced online data such as web searches provide better predictions on human migration than existing models, we decided to compare 3 different models following diverse metrics.

- The first model will be based on existing migration techniques from the literature and will be used as benchmark. As Böhme et al. [9] already showed that adding Google Trends data to traditional approaches like the gravitational model improves the predictive power, we use a reproduction of their bilateral model relying on linear regression (see equation 3.5) as a benchmark prototype to beat.

- Secondly we use a basic machine learning model to verify the assumption that machine learning techniques are relevant in the scope of forecasting human migrations. We choose a basic ANN model as Robinson and Dilkina [60] presents very promising and encouraging results for it in this domain.

- Finally we develop a more complex deep learning technique for the third model: the Long Short-Term Memory (LSTM) which is a Recurrent Neural Networks (RNN) architecture. The purpose is to show that this last model yields better results than the two others.

The source code used to define, validate and evaluate the presented approaches is available on the following git repository:

`https://github.com/aia-uclouvain/gti-mig-paper`

It contains the script to extract the Google Trends Index, the *Google Colab* notebook to run the different experiments, as well as the data we used. The code is written in python and uses the *Keras* library, which runs on top of TensorFlow to build the two machine learning models while the *scikit-learn* library is used for the OLS, for the standardisation of the data and to compute certain metrics.

4.1 Data

We roughly employ the same set of data as Böhme et al. [9] used for the bilateral model (equation 3.5) since we choose that model to be our benchmark. This choice is convenient for a variety of reasons: (a) it allows to ensure that our reproduction of their model is faithful by comparing our results with the one presented in the paper; (b) since their data is available online, it allows us to avoid the tedious extraction of GTIs due to the GT rate limits (see Appendix C); and finally (c) it allows us to show that for exactly the same set of values they used, there exist better performing models.

As explained in sections 2.3 and 3.3, when using machine learning techniques, the idea is to fit f when predicting $\hat{Y} = f(features)$. As a reminder, for our application the dependent variable Y is the next year annual migration flow $T_{i,j,t+1}$ between a country of origin i and a country of destination j while the $features$ are geo-referenced demographic, economic and online search data about the countries of origin and destination for the current year t.

The migration data are provided by the OECD [54]. It provides a yearly bilateral migratory flow from 101 countries of origin to the 35 countries member of the OECD from the early 2004 until 2015 which makes a total of 23 947 observations. It is important to notice that this data is very sparse and that there are a lot of missing values: for a majority of origin-destination pair of countries, the values of annual migration flows are available of only a subset of the years from 2004 to 2015. Furthermore, a majority of the annual migration flows are near zero observations and there are very few extremely important ones but we will discuss about it later in section 5.1 in a deeper analysis.

Table 4.1 gives an overview of the different features used in the three models. From the set of control variables provided by Böhme et al. [9] (see section 3.2.1), we choose the following factors: Gross Domestic Product (GDP) and population size for both origin and destination countries. Since the extended variables presented in section 3.2.1 are not available for every country and since the purpose of this thesis is to analyse the predictive power of online search data, we limit these economic and demographic factors as much as possible while keeping common and representative indicators about the size and the economy of the country.

To limit the number of features without impacting too severely the model's performance, we restrained the fixed-effects to three one-hot vectors representing the

Tab. 4.1.: The output variable along with the input features used for the different models. Each feature spans from 2004 to 2014 for a pair of origin-destination country. Refer to subsection 3.1 for a detailed explanation of the Google Trends Index (GTI).

Dependent variable	**Description**
$T_{i,j,t+1}$	Next year migration flow from country i to country j
Input features$_{i,j,t}$	**Description**
$GDP_{i,t}$	Gross Domestic Product for origin country i during the year t
$GDP_{j,t}$	Gross Domestic Product for destination country j during the year t
$pop_{i,t}$	Population size for origin country i during year t
$pop_{j,t}$	Population size for destination country j during year t
$fixed_i$	Origin country i fixed effects, encoded as a one-hot vector
$fixed_j$	Destination country j fixed effects, encoded as a one-hot vector
$fixed_t$	Year t fixed effects, encoded as a one-hot vector
$GTIbi_{i,j,t}$	Bilateral GTI for a pair origin country i and destination country j during a year t
$GTIuni_{i,t} \times GTIdest_{i,j,t}$	Unilateral and destination GTI for an origin country i, a destination country j during a year t
$T_{i,j,t}$	Current year migration flow from country i to country j

concerned country of origin ($fixed_i$), the concerned country of destination ($fixed_j$) and the concerned year ($fixed_t$).

We could use a single one-hot vector $fixed_{i,j}$ specific to the pair of origin and destination country i, j as it would allow to absorb time-invariant factors like the distance between the 2 countries or the presence of common language, etc. However the dimension of the vector would significantly increase as we need an entry for each pair of origin and destination countries instead of an entry for all origin and destination countries[1]. The low number of observations pushes us to restrict the size of input vectors in order to sustain an acceptable ratio of number of features per observation and to limit overfitting [30].

Finally we added a last feature $T_{i,j,t}$ not initially present in the bilateral model which is the value of the migration flow for the current year because it provides a strong additional predictive power.

Obviously we keep the different forms of the bilateral GTI: $GTIbi_{i,j,t}$ and $GTIuni_{i,t} \times GTIdest_{i,j,t}$.

Since the Google Trends data start in 2004, that the migration data used by Böhme et al. [9] stops in 2015 and that we want to predict the next year migration, the input features data spans from 2004 to 2014 and the output variable spans from 2005 to 2015. That means that we have time-series of migration flows between 2 countries of maximum 11 years in length. Moreover we decide to drop the observations for pair of origin-destination countries where migration data was available for only one year, that is, the time-series of length 1, because these observations were not relevant

[1] Since we have 101 countries of origin and 35 countries of destination the dimension ratio between $fixed_{i,j}$ and $fixed_i \cup fixed_j$ is $\frac{101*35}{101+35} \approx 26$.

(almost always near zero migration flows) and biased heavily the performance of the different models due to the lack of data for these pairs of countries.

To conclude it leaves us with 101 countries of origin, 34 countries of destination, 1997 time-series of length in the range from 2 to 11 for a total of 19 326 observations.

Thereafter, we will need to divide our data into 3 different sets:

- a training set which will be used to fit our models;
- a validation set with which we will not use to fit our models in the first place because it will be used to regulate the hyper-parameters. We assess the models with different parameters on this set of data to determine which ones lead to the best performances;
- a test set which will be used to evaluate the final predictive power of each model once all their parameters are fixed.

As we want to evaluate the capacity of the models to predict migration flows of future years, we divide the initial set of data by subgroup of years. The training set gathers the input features from 2004 to 2012 (input features$_{i,j,04..12}$ $\forall i, j$) and also all the observed migration flows spanning from 2005 to 2013 as output (T$_{i,j,05..13}$ $\forall i, j$) since we predict next year migration. The validation set consists of input features on the year 2013 (input features$_{i,j,13}$ $\forall i, j$) and migration flows of 2014 (T$_{i,j,14}$ $\forall i, j$) while the test set captures output migration flows of the year 2015 (T$_{i,j,15}$ $\forall i, j$) and input values from 2014 (input features$_{i,j,14}$ $\forall i, j$).

Since the validation and the test sets each gathers data for one of the 11 years available, each of these sets represents slightly less than 10% of the whole data.

It is important to clarify that when we assess the performances of our models on the test set, that is, after the validation on hyper-parameters, we fit the models both on the training and validation set.

Metrics

The performance of the prediction models can be evaluated with several metrics. We present below the metrics used in [60] as well as their equations.

Common Part of Commuters (CPC) : It is a common metric used in the literature to evaluate human mobility models [46, 60]. Its value is 0 when the ground matrix $\mathbf{T}$ and the prediction matrix $\hat{\mathbf{T}}$ have no entries in common, and 1 when they are identical:

$$CPC(\mathbf{T}, \hat{\mathbf{T}}) = \frac{2\sum_{i,j=1}^{m,n} \min\left(T_{i,j}, \hat{T}_{i,j}\right)}{\sum_{i,j=1}^{m,n} T_{i,j} + \sum_{i,j=1}^{m,n} \hat{T}_{i,j}} \tag{4.1}$$

Mean Absolute Error (MAE) : A standard and widely used measure, its value is 0 when the values of both matrices are identical, and arbitrarily positive the worse the prediction gets:

$$MAE(\mathbf{T}, \hat{\mathbf{T}}) = \frac{1}{m \cdot n} \sum_{i,j=1}^{m,n} |T_{i,j} - \hat{T}_{i,j}| \tag{4.2}$$

Root Mean Square Error ($RMSE$) : Another standard measure to evaluate predictions, its value is 0 when the values of both matrices are identical, and arbitrarily positive the worse the prediction gets. The main difference with the MAE is that the RMSE penalises more strongly the large errors:

$$RMSE(\mathbf{T}, \hat{\mathbf{T}}) = \sqrt{\frac{1}{m \cdot n} \sum_{i,j=1}^{m,n} (T_{i,j} - \hat{T}_{i,j})^2} \tag{4.3}$$

Coefficient of determination (r^2) : It measures the goodness of fit between the predictions and the ground truth values. Its value is 1 when the predictions perfectly fits the ground truth values, 0 when the predictions are identical to the expectation of the ground truth values, and arbitrarily negative the worse the fit gets:

$$r^2(\mathbf{T}, \hat{\mathbf{T}}) = 1 - \frac{\sum_{i,j=1}^{m,n} (T_{i,j} - \hat{T}_{i,j})^2}{\sum_{i,j=1}^{m,n} (T_{i,j} - \overline{T})^2} \tag{4.4}$$

Mean Absolute Error In (MAE_{in}) : That is, the MAE on total incoming migrant by destination countries $v_j = \sum_{i=1}^{m} T_{i,j}$. Following the argument given by Robinson and Dilkina [60], we justify this choice of metric because even if the total incoming migrants in a country provides less information compared to the amounts migrants from each origin country (to this destination country),

it remains a very important and more direct measure for policy makers as it can predict future population growth:

$$MAE_{in}(\mathbf{v}, \hat{\mathbf{v}}) = \frac{1}{n}\sum_{j}^{n} |v_j - \hat{v}_j| \tag{4.5}$$

In order to make fair comparisons, for our experiments, we use these metrics.

4.2 Gravity Approach

As explained earlier, we use a reproduction of the bilateral gravity model estimated through an OLS regression from Böhme et al. [9] as benchmark model. Since the authors presented different versions of the model, we choose the most straightforward form using the Google Trends data whose simplified base 10 logarithm gravity equation is represented below[2].

$$\begin{aligned} \log T_{i,j,t+1} = {} & \beta_1\ GTIbi_{i,j,t} + \beta_2\ GTIuni_{i,t} \times GTIdest_{i,j,t} + \\ & \beta_3\ \log GDP_{i,t} + \beta_4\ \log pop_{i,t} + \beta_5\ \log GDP_{j,t} + \\ & \beta_6\ \log pop_{j,t} + fixed_i + fixed_j + fixed_t + \epsilon_{i,j,t} \end{aligned} \tag{4.6}$$

By comparing the above equation with the complete version of the gravity's equation described in 3.5, one can notice that we restrain the control variables to the GDP and the population size of the origin and destination countries and we use 3 simplified vectors of fixed-effects encoded as one-hot vectors.

To ensure that our reproduction of the gravity type model proposed by Böhme et al. [9] is faithful, we compare the linear regression's coefficients and the r^2-score that we obtain after a fit on the whole dataset with theirs and we find very similar results. The analysis of these coefficients allow us to observe the impact of the different features on the predictions of this model.

Tab. 4.2.: The coefficients of the different control variables obtained by the fit of the gravity model 3.5 estimated through an OLS regression on the whole data.

Feature	$\log GDP_{i,t}$	$\log pop_{i,t}$	$\log GDP_{j,t}$	$\log pop_{j,t}$
Coefficient	-0.057	1.299	-0.0079	-0.212

[2] Once again, the logarithm of the following equation are computed on the value of the variable **plus 1** in order to avoid any complications with null values.

We represent in Table 4.2 the coefficients of the demographic and economic features. As suspected, the population size of the origin country has a strong positive impact on the prediction of the migration flows. On the other hand, the population size of the destination country surprisingly has a negative impact on it. These coefficients tend to show that the GDP has significantly less influence on the estimation compared to the population. However, these surprising results can be partially explained by the fact that the vectors of fixed-effects already absorb time-invariant demographic and economic factors at the origin and destination countries level.

By analysing the coefficients corresponding to the GTIs, we notice that the first type of bilateral GTIs ($GTIbi_{i,j,t}$) have a mean absolute weight of 0.03 while the second type of bilateral GTIs ($GTIuni_{ot} \times GTIdest_{odt}$) have a mean absolute weight of 0.000 15[3]. The model grants greater importance to the first type of GTIs although they are much more sparse (94% of null values) than the second type (34% of null values). This shows that the $GTIbi_{i,j,t}$ better represent the intentions of migration to the destination country d than the $GTIuni_{i,t}$ multiplied by the $GTIdest_{i,j,t}$.

The weights assigned to the one-hot vectors of origin and destination countries ($fixed_i$ and $fixed_j$) have respective mean absolute values of 1.51 and 1.23. This shows that these fixed-effects play a very important role in the estimation of migratory flows. The coefficients corresponding to the destination fixed-effects are approximately proportional to the population size: the highest weight is associated to the USA (3.44) while the lowest is associated to the Latvia (-2.91). This tendency is not observed with the origin countries: there is almost no correlation between the population size of the country and the associated weight. This confirms our hypothesis: the value of the coefficient linked to the population size of the destination country is counter-intuitive because the destination fixed-effect already absorbs this information.

Finally, by observing the linear regression's coefficients corresponding to the years of migration presented in Table 4.3, we can observe a general tendency to increase with time. We can conclude that with the exception of certain years, the more the years advance, the more the migratory flows of our database are important.

[3]Notice that to best match the original model no data standardisation has been performed. This implies that the $GTIbi_{i,j,t}$ are spanning from 0 to 100 while the $GTIuni_{i,t} \times GTIdest_{i,j,t}$ range from 0 to 10 000. It explains why the second ones have such low coefficients. However, their weights remain proportionally lower since the ratio between the two means is 1/200.

Tab. 4.3.: The coefficients associated to the different years of the one-hot vector $fixed_t$ obtained by the fit of the gravity model 3.5 estimated through an OLS regression on the whole data.

Year	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014
Coefficient	-0.2	-0.17	-0.15	0.0	0.01	0.07	-0.04	0.04	0.11	0.11	0.2

The various analysis that we carried out highlight an advantage that this first model has compared to the following. Even if it turns out that linear regression is less efficient than machine learning techniques, it nevertheless remains much more interpretable than these latter. Indeed, such studies on the impact of the different features is much more complex or even impossible with the ANN and LSTM models.

4.3 ANN Approach

Our second model is a deep learning based artificial neural network (ANN) model as proposed in Robinson and Dilkina [60]. Our ANN is composed of densely connected with rectified linear units (ReLU) activation layers and use a *Adam* optimiser (refer to Chapter 3). As presented in Figure 4.1, we use the same model for all the predictions with a time-step of 1 year. This means that the ANN receives as input the set of features $input\ features_{i,j,t}$ described in Table 4.1 and outputs the predicted next-year migration flow $T_{i,j,t+1}$.

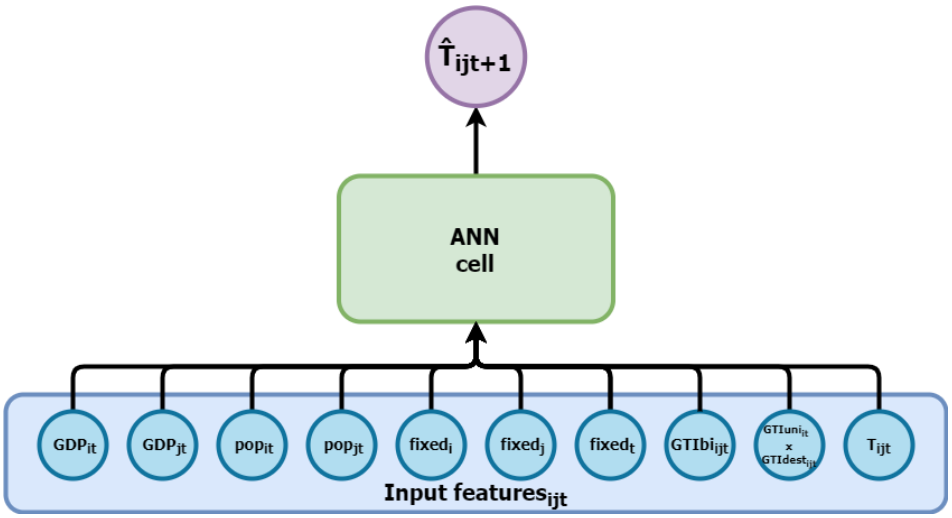

Fig. 4.1.: Architecture of the ANN approach.

Since it is a very common and recommended practice in machine learning and in regression analysis, we standardize the data in a range from 0 to 1 before being used by the neural network in order to improve the performance and to avoid obtaining misleading results . We use a *min-max scaler* [30] on each of the following variables:

$$\begin{aligned} &\forall i,t && GDP_{i,t} \;\&\; pop_{i,t} \\ &\forall j,t && GDP_{j,t} \;\&\; pop_{j,t} \\ &\forall i,j,t && GTIbi_{i,j,t} \;\&\; GTIuni_{i,t} \times GTIdest_{i,j,t} \;\&\; T_{i,j,t} \end{aligned}$$

That means for each of these 7 types of data, we apply the following transformation:

$$V = \frac{V - \min(V)}{\max(V) - \min(V)} \tag{4.7}$$

V being the set of all values of the concerned variable. Seeing that our data is very sparse for some features and that we do not have negative values, this scaler seems like an appropriate choice as it does not change null values.

Hyper-parameter Optimisation

The next step is to optimise our model by tuning different hyper-parameters. We validate the following hyper-parameters in this order: loss function, depth and width of hidden layers, training batch size, dropout and number of epochs.

First, we analyse the efficiency of the ANN approach depending on the associated loss function as presented in section 3.3.1: (a) MAE; (b) MSE; or (c) CPC.

Since the neural network's number (i.e., depth) and sizes (i.e, width) of hidden layers define the whole structure of the model, these are the parameters that most influence the quality of predictions. It is necessary to choose a judicial structure of ANN because a too complex model will lead to overfitting while a too simple structure will cause underfitting. See chapter 3 for more on that.

The batch size corresponds to the number of training samples propagated through the ANN before proceeding to the computation of the loss function, to the gradient descent and thus to the update of the internal model parameters. The larger the batch size, the fewer applications of the backpropagation algorithm, so the faster the model will be trained. However, too large values of batch size can impact negatively the prediction's quality. The objective is therefore to find a good compromise between speed and performance.

Adding a dropout regularisation in machine learning techniques is a common choice to reduce the overfitting and ensure a better generalisation [66, 24, 3]. It consists on a value ranged between 0 and 1 representing the proportion of random input features or hidden layers dropped before each passage trough the network.

Finally, the number of epochs defines the number of times the learning algorithm will be trained on the whole dataset. Nowadays, learning algorithms often need hundreds or even thousands of epochs to be completely trained, however to many epochs can cause the model to overfit [30].

For each of the previously selected hyper-parameters we proceed to the same validation method presented in the algorithm 1.

Algorithm 1: ANN Validation Algorithm

Data: *range*: list of parameter's values to be evaluated
Data: *epoch*: number of times the model is trained on the whole dataset
Result: The validation of hyper-parameter is done
for *each val in range* **do**
    model ← new_ANN_model(val) **for** *each epoch* **do**
        $\hat{T}_{05..13}$ ← *model* .fit(input_features$_{04..12}$, T$_{05..13}$)
        add_to_train_history(compute_metrics($T_{05..13}$,$\hat{T}_{05..13}$))
        $\hat{T}_{14}$ ← *model*.predict(input_features$_{13}$)
        add_to_valid_history(compute_metrics(T_{14},$\hat{T}_{14}$))
display_train_and_valid_histories

First, we choose a range of values $range$ for the concerned parameter. Then we create one model for each values val in this range. The models are defined with the optimised values for the already tuned hyper-parameters and with an arbitrary value for the others. Obviously, the currently tuned parameter is set to the corresponding value val. Afterwards, we train and evaluate the models: for each epoch, we fit them on the whole training set (see section 4.1) and then we calculate the values of the different metrics presented in section 4.1 on the predictions made on both the training and the validation sets. This allows us to observe for each previously defined model, the variations of the different metrics on the training and the validation sets depending on the number of the epoch.

We present some of these graphics obtained during the validation process in Appendix D.

Finally, after the validation on the different previously introduced hyper-parameters, we obtain the following optimal values: loss function - *MAE*, depth and width of

hidden layers - *2* layers of width *200*, training batch size - *32*, number of epochs - *170* and dropout - *0.1*.

4.4 LSTM Approach

Our third approach is based on a Recurrent Neural Network (RNN) composed of a single LSTM layer using an *Adam* optimiser in charge of predicting the bilateral flows $\hat{T}_{i,j,t}$. The particular architecture of a RNN allows to share features learned across different parts of the sequential data to persist through the network and it also does not require to have a fixed set of input features vectors.

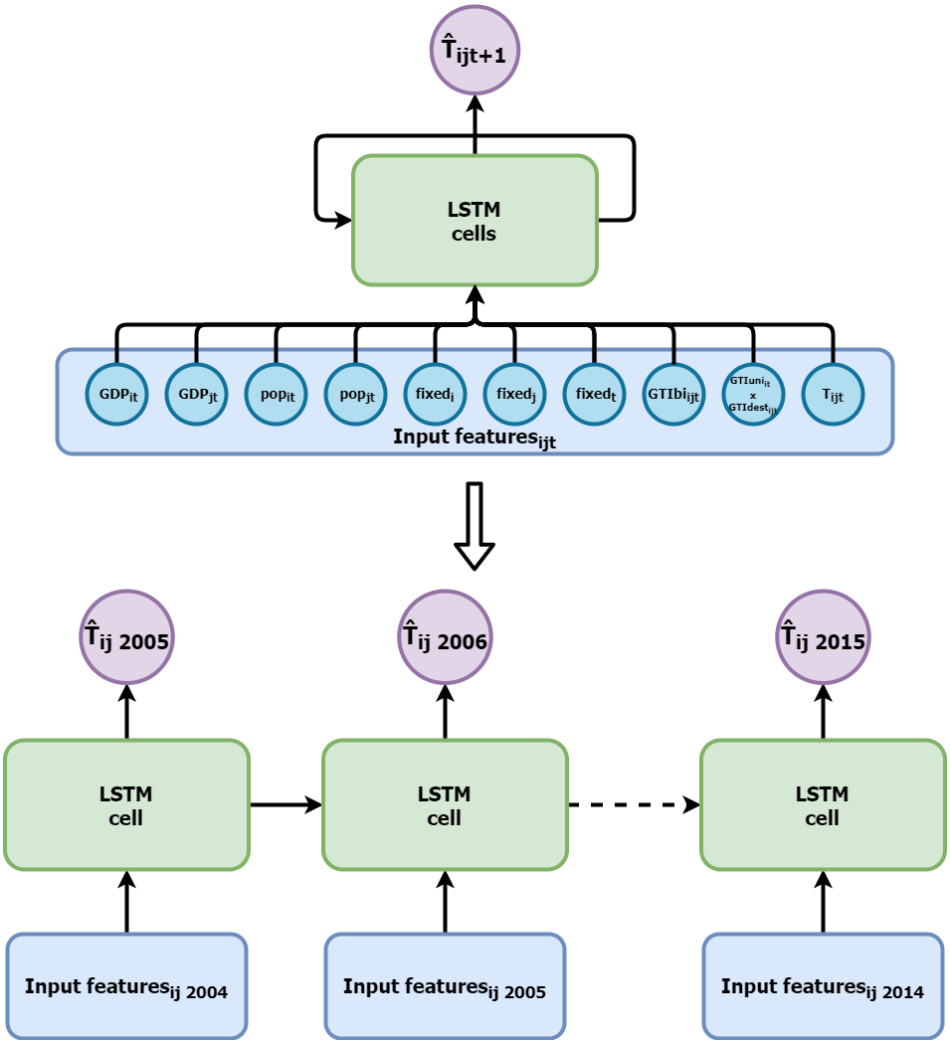

Fig. 4.2.: The unfolded gated-RNN with LSTM cells structure. The upper-side corresponds to the folded RNN, while the lower-side to the unfolded RNN.

As presented in Figure 4.2, the more flexible nature of RNN allows us to exploit the time-related relations of our data by fitting the sequences of LSTM cells with our origin-destination time series $T_{i,j,04..15}$[4]. Since our time series are not all of the same length[5], it is very advantageous to be able to use non-fixed size LSTM cell sequences.

Learning LSTM model

A simplified version of our LSTM training method is presented in Algorithm 2, while our LSTM evaluation process is presented in Algorithm 3. Notice that the span of years presented in the algorithms corresponds to the one used once the validation is completed, that is, we fit our model on both the training and validation set.

Algorithm 2: LSTM Training Algorithm

Data: *model*: LSTM untrained model
Result: Model is trained
for *each epoch* **do**
  for *each pair i,j of origin-destination countries* **do**
    `/* gradient descent for each batch: */`
    model.fit(input_features$_{i,j,04..13}$, T$_{i,j,05..14}$)
evaluation(*model*) `/* see algorithm 3 */`

Algorithm 3: LSTM Evaluation Algorithm

Data: *model*: LSTM trained model
Result: Model is evaluated
for *each pair i,j of origin-destination countries* **do**
  $\hat{T}_{i,j,05..15} \leftarrow$ *model*.predict(input_features$_{i,j,04..14}$)
  train_error $\leftarrow$ compute_metrics($T_{05..14}$,$\hat{T}_{05..14}$)
  test_error $\leftarrow$ compute_metrics(T_{15},$\hat{T}_{15}$)

Due to its specificity, we fit our LSTM time series by time series. Therefore we use a variable batch size corresponding to the number of years present in the serie . This implies that the gradient descent is applied and the LSTM's parameters are updated after each propagation of a time series through the LSTM cells (as presented in Figure 4.2).

[4] By time serie we mean the sequence of annual migration flows between a pair of origin destination countries i and j.
[5] As explain in section 4.1 about the data, for some pairs of origin and destination countries, they are missing bilateral migration flow data for certain years.

Furthermore the features are standardised by sequences of origin-destination in a range [0,1] using a min-max scaler. This means that for each pair i, j of origin and destination countries we apply the following normalisation:

$$V_{i,j} = \frac{V_{i,j} - \min(V_{i,j})}{\max(V_{i,j}) - \min(V_{i,j})} \quad \forall (i,j) \tag{4.8}$$

where $V_{i,j}$ is the set of values of one of the following standardised features for the specified countries i and j $\forall t$: $GDP_{i,t}$; $pop_{i,t}$; $GDP_{j,t}$, $pop_{j,t}$; $GTIbi_{i,j,t}$; $GTIuni_{i,t} \times GTIdest_{i,j,t}$; and $T_{i,j,t}$.

These modelling choices are well-considered for RNN working with multiple similar time sequences [15] and significantly improve the performance of our machine learning approach. Our LSTM model uses a bias of 1 for the LSTM forget gate since it has been shown to improve performances drastically [26, 42].

It is important to notice that we use the same LSTM model for all time series. Another approach would have been to use different networks to estimate the flows for each pair of countries. The amount of data to train each would have been very limited, though.

Hyper-parameter Optimisation

As in the ANN approach, we tune different LSTM hyper-parameters on the validation set in order to find their optimal values. We optimise the following hyper-parameters in this order: loss function, depth and size of hidden layers, dropout and number of epochs. Notice here that we do not validate the batch size parameter since it is already equal to the size of the different time series.

The validation process comes down to apply the training process presented in algorithm 2 with a different set of years (since the model learns only on the training set and makes predictions on the validation set) repeatedly by varying the value of the tuned hyper-parameter. Furthermore, an evaluation is performed on every epoch to assure a continuous assessment of the predictions quality depending on the number of epochs.

The validation graphs of the different parameters are available in Appendix E.

After the validation on the different previously introduced hyper-parameters, we obtain the following optimal values: loss function - *MAE*, number and size of hidden layers - *1* layer of width *50*, number of epochs - *50*, and dropout - *0.15*.

As for the ANN approach, the MAE loss function presents the best results during the validation which can be surprising since it does not penalise important errors like the MSE and it is not a metric used specifically to evaluate human mobility models like the CPC. However, as presented in the graphs (Appendix E), both the MAE and the CPC losses showed similar results in term of performance for the LSTM model with other hyper-parameters arbitrarily-fixed. This is why we have optimised the other hyper-parameters (number and size of hidden layers, dropout and number of epochs) both with a MAE and a CPC loss function to finally observe that the best model among all these was the one using the parameters values presented above.

5 Results and Discussion

We carry out experiments comparing the performance of the 3 different approaches described in section 4.

We fit the different models on both the training and validation sets and we use the test set gathering every migration flow taking place in 2015 as presented in section 4.1. It is important to understand that the training data used for the following experiments is thus composed of the training and validation sets.

5.1 Comparison of the Metrics Computed on the Different Approaches

We measure the different metrics mentioned in the Metrics section on the predicted migrations flows from the training data ($\hat{T}_{i,j,t} \quad \forall i,j \; \forall t \in [05..14]$) and on the predicted migrations flows from the test set ($\hat{T}_{i,j,15} \; \forall i,j$). The Table 5.1 shows the results thus obtained.

Tab. 5.1.: Comparison of the 3 models for the specified metrics. The values shown are by pair (train - test). Bold values indicate the best values per column. There are on average 742 annual migrants per bilateral migration flow and 46 119 annual incoming migrants per destination country.

	CPC		MAE		RMSE		r^2		MAE_{in}	
Models	train	test	train	test	train	test	train	test	train	test
Linear Regression	0.871	0.866	819	877	6 100	5 239	0.800	0.773	24 128	28 737
ANN	0.931	0.834	119	306	818	1 553	0.975	0.921	3 257	9 664
LSTM	**0.945**	**0.892**	**96**	**225**	**639**	**1 028**	**0.985**	**0.967**	**2 261**	**4 827**

We can observe that our ML models perform much better than the Böhme et al. [9]'s linear model. Indeed, with the same data, the ANN beats the first model in almost every metric while the LSTM model completely outperforms it in all the measures. The ANN model fits very well the training data but it does not seem to generalise as well as the LSTM model as shown by their performance on the test set. Of course, the results obtained on the test set are more indicative of the predictive power of each model because it represents its capacity to estimate unknown future migration

flows. We can draw from these first results that the LSTM is the best predictive model among these three.

Since the RMSE values are always way higher than the MAE (between 5 and 7 times larger) we can conclude that the models tend to make a few really large errors. This can be explained by analysing the data. In the dataset, the mean value of migration flows between 2 countries during a year is 742 but the median value is only 17 while the maximum is about 190 000. This indicates that our dataset is very sparse: there is a lot of near zero observations (40% are below 10) for a very few extremely important ones (less than 2% reach 10 000). One can notice that the mean absolute errors of the different models are very important compared to the mean annual migration flows (742 and 46 119, see Table 5.1 caption) but these values are heavily biased by the sparsity of the data and by the large errors made on the really important migration flows, e.g. the USA and Spain.

5.2 Difference between Truth Values and Estimations

The Figure 5.1 shows the ground truth values of every annual migration flows of the database ($T_{i,j,t}\ \forall i,j,t$) along with the associated values ($\hat{T}_{i,j,t}\ \forall i,j,t$) estimated by each model. This allows us to observe the nature of the errors realised by the different approaches. A little warning, these graphs represent without distinction the estimations resulting from the training set and the ones coming from the test set.

Furthermore, the green curves highlight the very sparse and uneven nature of the migration flow data.

The first graph exposes the fact that the gravity approach makes huge errors on important bilateral migration flows while the difference between the estimated and actual number of migrants is much smaller for near zero migrations. This implies that the mean results obtained in Table 5.1 are heavily biased by the small errors done on small bilateral migration flows. If we decide to compute the metrics on only interesting pairs of countries of origin and destination (let us say for example, the ones having more than 100 annual migrants) the increase of the resulting mean values would be significantly higher for this first approach than for the two others.

Machine learning approaches present estimations sticking much more to reality for large migration flows. That said we can see a slight tendency of the ANN model to underestimate the important migrations.

We can also observe that the second model outputs a high number of negative values. However these negative values appears for minor migration flows and a very small part of them are below -50. In the other hand, the LSTM approach graphic shows much less negative estimations but one of them reaches -700.

A first and very simple improvement for these two models would then be to automatically set negative estimations to 0.

5.3 Comparison of the Predictions from the Test Set

In order to have a better visualisation of the predictive power of the models, we represent in Figure 5.2 the scatter plot of the 3 models for the test set only. A scatter plot is a graph in which each axes represents the values of a certain variable. In our case, the ground truth values of the migrations flows from the test set $T_{i,j,15}\ \forall i,j$ are plotted along the x-axis while the estimated values $\hat{T}_{i,j,15}\ \forall i,j$ of the different models are represented along the y-axis. The predictions of each approach are associated with a different colour and the green curve expresses the perfect fit. So the closer the points are to it, the closer the prediction is to reality.

Again, the graph reflects well the sparse nature of the data as shown by the density of points along the x-axis.

As expected following the first results, we can observe that the linear model does not provide very accurate predictions. The ANN model, on the other hand, shows a stronger tendency to underestimate the ground truth values. Ultimately, the LSTM's estimations are the ones sticking the most to the actual migration flows which confirms our first assumption.

5.4 Comparison on the Total Incoming Migrants per Destination

Finally, we decide to study the results of the MAE_{in} metric in more details. As a reminder, this metric computes the absolute mean error on the total annual number of migrants moving each one of the 34 destination countries of the OECD (see section 4.1). These measures represent an important and direct indicator to predict future population growth so it deserves a deeper analysis.

In Figure 5.3 we show the error of the total number of incoming migrants per destination country per year for each model. The 34 destination countries from the OECD are represented on the vertical axis while the years of concerned migration flows are on the horizontal axis. We can observe that whatever the model, for the majority of countries and years, the estimation error is close to null and that the big errors often appear in the same countries of destination. Knowing that, we can see that the heatmaps of the ANN model and of the linear regression in Figure 5.3 highlight their tendency to underestimate the migration flows especially for the last year (the test year).

To compare these errors with the actual migration flows, we represent in the rightmost heatmap in Figure 5.3 the ground truth values of the total number of incoming migrants per destination country and per year in descending order. With this figure we can clearly see that the errors we make are mostly for the countries with important incoming migration flow.

In the case of Spain notice that there has been an important drop in incoming migration flows in 2008 due to the 2007–2008 financial crisis [16]. If we look at the LSTM model in Figure 5.3, we largely underestimate the predictions for Spain from 2005 to 2007. From 2008 and onwards, the prediction errors are comparatively smaller to those before that pivot year. This might indicate a lack of complexity of our model as it does not take into account major historical events like a financial crisis.

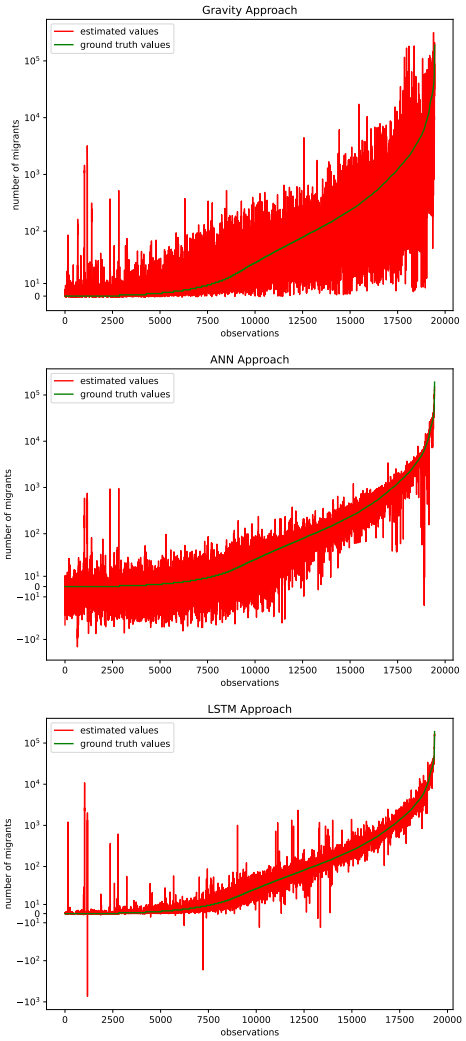

Fig. 5.1.: Plot of the difference between the migration flow $\hat{T}_{i,j,t}$ estimated by the different approaches and the actual one $T_{i,j,t}$ $\forall i, j, t$. The ground truth values are sorted in increasing order and represented in green while the associated estimated value is represented in red.

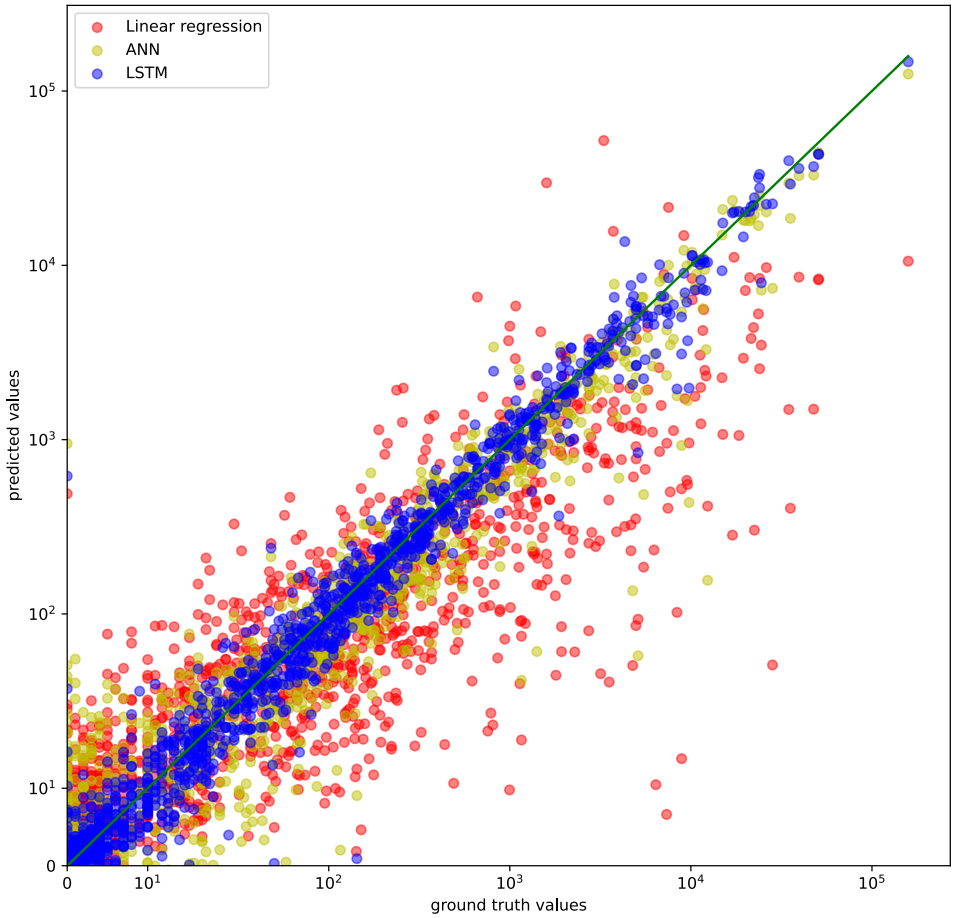

Fig. 5.2.: Scatter plot for the 3 models on the test set (year 2015) – The coefficient of determination for the linear regression is 0.773, for the ANN 0.921, and for the LSTM 0.967 (see the Table 5.1 for more details).

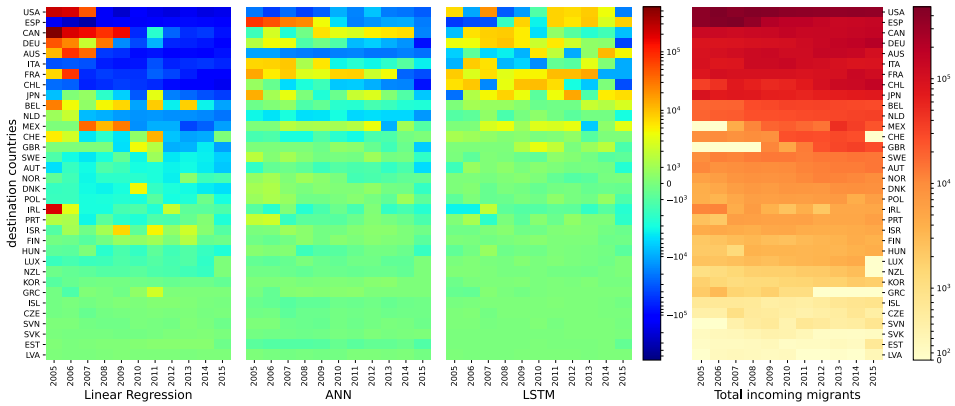

Fig. 5.3.: Heatmaps of the error on total incoming migrants for 34 OECD countries on the test year (2015) showing how well each model fits the data. From left to right: Linear Regression Model, ANN Model, and LSTM Model. The rightmost figure is the ground values for the total number of incoming migrants by destination countries. Countries are in descending order of total incoming migrants.

Conclusion

Böhme et al. [9] have recently demonstrated that including Google trends data in the set of standard features could improve the migration prediction models. In this work, relying exactly on the same data, we improved the quality of the prediction significantly by replacing the linear model used in by a Long short-term memory (LSTM) artificial recurrent neural network (RNN) architecture. Our experiments also demonstrated that the LSTM was outperforming a standard ANN on this task.

Limitations

One drawback of our machine learning approach is that we lose interpretability of the model and the predictions despite the high interpretability potential of Google search keywords. Machine learning techniques are very good at answering to the question: what is predicted? However, one of the key advantages of traditional models were their ability at answering to the question: how is it predicted?

Molnar [53] presents how to get the how part into the equation when using machine learning. For instance, the Local interpretable model-agnostic explanations (LIME) is a technique introduced by [59] to shed some light on the machine learning black box. It thus so by building a simple interpretable model near the prediction and is able to explain the output of any model. Further one, Karpathy et al. [43] shows how LSTM can be visualised even for text and tabular data.

In this work we constrained ourselves to use an already fixed set of keywords to extract the GTI. It uses a very straightforward and lax method of feature selection based on the most semantically related keywords. The drawback from this method is that the bilateral Google Trends Index feature are very sparse and contain 94% of 0s. A better way would have been to use some feature selection as insisted by Molnar [53]. One way could be to use a decision tree model like XGBoost [10]. In deed, decision trees classify the features according to their importance and might give some intelligence about which keywords ought to be kept or dropped. A more common technique is the Principal Component Analysis (PCA) which is a technique

to reduce the number of features in a dataset. Again, it will classify the features but this time according to the eigenvalues of the covariance matrix of the input set.

In the previously presented approaches we restrained the economic and demographic features to the population size and the GDP for diverse reasons. Firstly because the main purpose was to study the predictive power of Google Trends data but also because it is complex to obtain these additional features for every origin country. However the results have shown that the models lacks some information to acknowledge important variation of number of incoming migrants in some destination countries. Adding some factors like the presence of catastrophic events (financial crisis, war, natural disaster), unemployment, share of internet users, etc. could significantly improve our approaches.

Moreover, as explained in section 4.1, we use different vectors of fixed effects for the origin and destination countries to maintain an acceptable ratio number of observations per feature. The quantity of observed migration flow $T_{i,j,t}$ is mainly restrain by the span of years available (2004–2015). Nonetheless, the amount of migration data will increase over time and there are other ways to address this lack of data. So the use of a single vector of fixed effect for both the origin and destination country $fixed_{i,j}$ absorbing more complex time-invariant factors like the distance between the 2 countries, the presence of common language, etc. becomes much more reasonable.

Perspectives

As we said earlier, one of the main use of these kinds of migration models is to nowcast. High-performance models like the LSTM approach we presented can be used to fill in the gaps of migration data created by the lack of information, the low frequency or the long lag of publishing data. One could then complete the missing values from the OECD's migration data by the estimated values returned by the LSTM approach and then retrain a new model with the completed data. The same process could be applied to add more recent years to the actual database. We could iteratively repeat the process to add migration data year by year.

In the same order of ideas, a new version of the approaches using only Google Trends data could be explored. This would allow to benefit of the high frequency nature of GTI and use monthly time series.

We considered another method of validation and evaluation of the LSTM approach which would have allowed a more detailed analysis of its performances but this was too time consuming. Instead of defining fixed sets of training, validation and test, using sets of increasing-size inspired by the *sliding window* technique. The first training set would be only composed of data dating from the first available year ($T_{i,j,05}$) and the validation and test sets would gather data on the next two years (respectively $T_{i,j,06}$ and $T_{i,j,07}$). Then we would add iteratively one year to the training set ($T_{i,j,05..06}$, $T_{i,j,05..07}$) and slide the two others sets to one year later. This would allow us to assess the LSTM depending on the number of years on which it learns and thus this would give an idea of the performance of the model on future years.

As future work, we would like also to apply the latest interpretability techniques (see [53]) to better identify what the most important features are for making high quality migration predictions. This would equip economists, demographs and experts in migration with new tools to shed light on migration mechanisms.

Bibliography

[1] Mohammed N Ahmed, Gianni Barlacchi, Stefano Braghin, et al. “A Multi-Scale Approach to Data-Driven Mass Migration Analysis”. In: *SoGood@ ECML-PKDD* (2016), p. 17 (cit. on pp. 1, 8).

[2] James E. Anderson. “The Gravity Model”. en. In: *Annual Review of Economics* 3.1 (Sept. 2011), pp. 133–160 (cit. on p. 8).

[3] Devansh Arpit, Stanislaw Jastrzebski, Nicolas Ballas, et al. “A Closer Look at Memorization in Deep Networks”. In: (July 1, 2017). arXiv: `1706.05394 [cs, stat]` (cit. on p. 36).

[4] Nikolaos Askitas and Klaus F. Zimmermann. “The Internet as a Data Source for Advancement in Social Sciences”. In: *International Journal of Manpower* 36.1 (2015), pp. 2–12 (cit. on pp. 1, 7, 8).

[5] Nikos Askitas and Klaus F. Zimmermann. *Google Econometrics and Unemployment Forecasting*. en. SSRN Scholarly Paper ID 1465341. Rochester, NY: Social Science Research Network, May 2009 (cit. on p. 8).

[6] Victoria Bell. *How Much Time Does YOUR Country Spend Online?* Feb. 1, 2019. URL: `https://www.dailymail.co.uk/sciencetech/article-6658237/How-time-does-country-spend-online.html` (visited on Apr. 27, 2020) (cit. on p. 7).

[7] Yoshua Bengio, Patrice Simard, and Paolo Frasconi. “Learning Long-Term Dependencies with Gradient Descent Is Difficult”. In: *IEEE transactions on neural networks* 5.2 (1994), pp. 157–166 (cit. on p. 25).

[8] Johanna Blomqvist. *Using XGBoost to Classify theBeihang Keystroke Dynamics Database*. Uppsala universitet, Datalogi, 2018 (cit. on p. 10).

[9] Marcus H. Böhme, André Gröger, and Tobias Stöhr. “Searching for a Better Life: Predicting International Migration with Online Search Keywords”. In: *Journal of Development Economics*. Special Issue on Papers from “10th AFD-World Bank Development Conference Held at CERDI, Clermont-Ferrand, on June 30 - July 1, 2017” 142 (Jan. 1, 2020), p. 102347 (cit. on pp. 1, 8, 13, 15, 16, 19, 27–29, 32, 41, 47, 57, 61, 62, 64).

[10] Tianqi Chen. “Introduction to Boosted Trees”. In: *University of Washington Computer Science* 22 (2014), p. 115 (cit. on pp. 10, 47).

[11] Tianqi Chen and Carlos Guestrin. “Xgboost: A Scalable Tree Boosting System”. In: *Proceedings of the 22nd Acm Sigkdd International Conference on Knowledge Discovery and Data Mining*. 2016, pp. 785–794 (cit. on p. 10).

[12] Hyunyoung Choi and Hal Varian. “Predicting the Present with Google Trends”. In: *Economic Record* 88.s1 (2012), pp. 2–9 (cit. on pp. 1, 7).

[13] François Chollet et al. “Keras”. In: (2015) (cit. on p. 26).

[14] Francesco D’Amuri and Juri Marcucci. “The Predictive Power of Google Searches in Forecasting US Unemployment”. en. In: *International Journal of Forecasting* 33.4 (Oct. 2017), pp. 801–816 (cit. on p. 8).

[15] *Deep Learning for Time Series Forecasting*. en-US (cit. on p. 39).

[16] Andreu Domingo. “El Sistema Migratorio Hispano-Americano del Siglo XXI México y España”. es. In: *Revista de Ciencas y Humanidades - Fundación Ramón Areces* (Dec. 2017) (cit. on p. 44).

[17] Kenji Doya. “Bifurcations of Recurrent Neural Networks in Gradient Descent Learning”. In: *IEEE Transactions on neural networks* 1.75 (1993), p. 218 (cit. on p. 25).

[18] David M. Eberhard, F. Simons Gary, and D. Fennig Charles. *Ethnologue: Languages of the World*. 2020. URL: `https://www.ethnologue.com/` (visited on Apr. 11, 2020) (cit. on p. 14).

[19] Liran Einav and Jonathan Levin. “Economics in the Age of Big Data”. In: *Science* 346.6210 (Nov. 7, 2014). pmid: 25378629 (cit. on p. 7).

[20] *EM-DAT | The International Disasters Database*. https://www.emdat.be/ (cit. on p. 18).

[21] Robin Flowerdew and Murray Aitkin. “A Method of Fitting the Gravity Model Based on the Poisson Distribution”. In: *Journal of regional science* 22.2 (1982), pp. 191–202 (cit. on p. 9).

[22] Y. Fondeur and F. Karamé. “Can Google Data Help Predict French Youth Unemployment?” en. In: *Economic Modelling* 30 (Jan. 2013), pp. 117–125 (cit. on p. 8).

[23] Jerome H. Friedman. “Greedy Function Approximation: A Gradient Boosting Machine”. In: *Annals of statistics* (2001), pp. 1189–1232 (cit. on p. 10).

[24] Yarin Gal and Zoubin Ghahramani. “A Theoretically Grounded Application of Dropout in Recurrent Neural Networks”. In: (Oct. 5, 2016). arXiv: 1512.05287 [stat] (cit. on pp. 26, 36).

[25] General Mills. *GeneralMills/Pytrends*. General Mills, Oct. 29, 2019 (cit. on pp. 13, 63).

[26] Felix A. Gers, Jürgen Schmidhuber, and Fred Cummins. “Learning to Forget: Continual Prediction with LSTM”. In: *Neural Computation* 12.10 (Oct. 2000), pp. 2451–2471 (cit. on pp. 25, 26, 39).

[27] Felix A. Gers, Nicol N. Schraudolph, and Jürgen Schmidhuber. “Learning Precise Timing with LSTM Recurrent Networks”. In: *Journal of machine learning research* 3 (Aug 2002), pp. 115–143 (cit. on p. 26).

[28] C Lee Giles. “Noisy Time Series Prediction Using Recurrent Neural Networks and Grammatical Inference”. In: *Machine learning* 44.1-2 (2001), pp. 161–183 (cit. on p. 25).

[29] Jeremy Ginsberg, Matthew H. Mohebbi, Rajan S. Patel, et al. “Detecting Influenza Epidemics Using Search Engine Query Data”. en. In: *Nature* 457.7232 (Feb. 2009), pp. 1012–1014 (cit. on pp. 1, 7).

[30] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT press, 2016 (cit. on pp. 21, 23–26, 29, 35, 36).

[31] Google. *FAQ about Google Trends Data - Trends Help*. `https://support.google.com/trends/answer/4365533?hl=en&ref_topic=6248052`. 2020 (cit. on p. 14).

[32] Google. *Google Trends*. 2020. URL: `https://www.google.com/trends` (visited on Apr. 2, 2020) (cit. on pp. 13, 14).

[33] *Google Search Statistics - Internet Live Stats*. URL: `https://www.internetlivestats.com/google-search-statistics/#trend` (visited on Apr. 27, 2020) (cit. on p. 7).

[34] Alex Graves. *Supervised Sequence Labelling with Recurrent Neural Networks*. Vol. 385. Studies in Computational Intelligence. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012 (cit. on pp. 24, 25).

[35] Klaus Greff, Rupesh Kumar Srivastava, Jan Koutník, Bas R. Steunebrink, and Jürgen Schmidhuber. “LSTM: A Search Space Odyssey”. In: *IEEE Transactions on Neural Networks and Learning Systems* 28.10 (Oct. 2017), pp. 2222–2232. arXiv: 1503.04069 (cit. on pp. 25, 26).

[36] Sepp Hochreiter. “Untersuchungen Zu Dynamischen Neuronalen Netzen”. In: *Diploma, Technische Universität München* 91.1 (1991) (cit. on p. 25).

[37] Sepp Hochreiter and Jürgen Schmidhuber. “Long Short-Term Memory”. In: *Neural computation* 9.8 (1997), pp. 1735–1780 (cit. on pp. 1, 25).

[38] International Displacement Monitoring Center IDMC. *Global Report on Internal Displacement*. Apr. 2020 (cit. on p. 5).

[39] International Organization for Migration IOM. *Global Migration Trends*. Oct. 30, 2018. URL: `https://www.iom.int/global-migration-trends` (visited on Oct. 4, 2019) (cit. on p. 6).

[40] International Organization for Migration IOM. *Glossary on Migration*. Vol. International Migration Law. Alice Sironi, Céline Bauloz and Milen Emmanuel. 2019 (cit. on p. 5).

[41] International Organization for Migration IOM. *World Migration Report 2020*. 2019 (cit. on pp. 5, 6).

[42] Rafal Jozefowicz, Wojciech Zaremba, and Ilya Sutskever. “An Empirical Exploration of Recurrent Network Architectures”. In: *International Conference on Machine Learning*. 2015, pp. 2342–2350 (cit. on pp. 25, 26, 39).

[43] Andrej Karpathy, Justin Johnson, and Li Fei-Fei. "Visualizing and Understanding Recurrent Networks". In: *arXiv:1506.02078 [cs]* (Nov. 2015). arXiv: 1506.02078 [cs] (cit. on p. 47).

[44] Simon Kemp. *Digital 2019: Global Internet Use Accelerates*. Jan. 30, 2019. URL: https://wearesocial.com/blog/2019/01/digital-2019-global-internet-use-accelerates (visited on Apr. 27, 2020) (cit. on p. 7).

[45] Diederik P. Kingma and Jimmy Ba. "Adam: A Method for Stochastic Optimization". In: *arXiv preprint arXiv:1412.6980* (2014) (cit. on p. 22).

[46] Maxime Lenormand, Sylvie Huet, Floriana Gargiulo, and Guillaume Deffuant. "A Universal Model of Commuting Networks". In: *PloS one* 7.10 (2012) (cit. on pp. 11, 31).

[47] Emmanuel Letouzé, Mark Purser, Francisco Rodríguez, and Matthew Cummins. "Revisiting the Migration-Development Nexus: A Gravity Model Approach". In: *Human Development Research Paper 44* (2009) (cit. on p. 9).

[48] Hao Liang, Meng Zhang, and Hailan Wang. "A Neural Network Model for Wildfire Scale Prediction Using Meteorological Factors". In: *IEEE Access* 7 (2019), pp. 176746–176755 (cit. on pp. 20, 25).

[49] Monty G Marshall and Gabrielle Elzinga-Marshall. "TABLE 1: STATE FRAGILITY INDEX AND MATRIX 2016". en. In: (2016), p. 10 (cit. on p. 18).

[50] A. Paolo Masucci, Joan Serras, Anders Johansson, and Michael Batty. "Gravity versus Radiation Models: On the Importance of Scale and Heterogeneity in Commuting Flows". In: *Physical Review E* 88.2 (2013), p. 022812 (cit. on pp. 1, 9, 10).

[51] Jacques Melitz and Farid Toubal. "Native Language, Spoken Language, Translation and Trade". en. In: *Journal of International Economics* 93.2 (July 2014), pp. 351–363 (cit. on p. 18).

[52] *Migration and Its Effects: Causes, Migrants, Impacts, Videos & Questions*. URL: https://www.toppr.com/guides/evs/no-place-for-us/migration-and-its-effects/ (visited on Apr. 27, 2020) (cit. on p. 14).

[53] Christoph Molnar. *Interpretable Machine Learning*. 2019 (cit. on pp. 47, 49).

[54] OECD. *International Migration Database*. 2020. URL: https://www.oecd-ilibrary.org/content/data/data-00342-en (cit. on pp. 5, 6, 15, 17, 28, 63).

[55] Christopher Olah. *Understanding LSTM Networks – Colah's Blog*. Aug. 27, 2015. URL: http://colah.github.io/posts/2015-08-Understanding-LSTMs/ (visited on Jan. 30, 2020) (cit. on p. 26).

[56] Jacques Poot, Omoniyi Alimi, Michael P Cameron, and David C Maré. "The Gravity Model of Migration: The Successful Comeback of an Ageing Superstar in Regional Science". en. In: (2016), p. 27 (cit. on p. 9).

[57] Ernest George Ravenstein. "The Laws of Migration". In: *Journal of the royal statistical society* 52.2 (1889), pp. 241–305 (cit. on p. 8).

[58] Ernst Georg Ravenstein. “The Laws of Migration”. In: *Journal of the statistical society of London* 48.2 (1885), pp. 167–235 (cit. on p. 8).

[59] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. “" Why Should i Trust You?" Explaining the Predictions of Any Classifier”. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2016, pp. 1135–1144 (cit. on p. 47).

[60] Caleb Robinson and Bistra Dilkina. “A Machine Learning Approach to Modeling Human Migration”. In: *Proceedings of the 1st ACM SIGCAS Conference on Computing and Sustainable Societies*. 2018, pp. 1–8 (cit. on pp. 10, 20, 22, 23, 27, 30, 31, 34).

[61] Simon Rogers. *What Is Google Trends Data — and What Does It Mean?* en. `https://medium.com/google-news-lab/what-is-google-trends-data-and-what-does-it-mean-b48f07342ee8`. July 2016 (cit. on p. 14).

[62] James M. Rubenstein. *Cultural Landscape, The: An Introduction to Human Geography*. en. /content/one-dot-com/one-dot-com/us/en/higher-education/product.html (cit. on p. 14).

[63] Sebastian Ruder. “An Overview of Gradient Descent Optimization Algorithms”. In: *arXiv preprint arXiv:1609.04747* (2016) (cit. on p. 22).

[64] Jürgen Schmidhuber, Daan Wierstra, and Faustino J. Gomez. “Evolino: Hybrid Neuroevolution/Optimal Linear Search for Sequence Prediction”. In: *Proceedings of the 19th International Joint Conferenceon Artificial Intelligence (IJCAI)*. 2005 (cit. on p. 25).

[65] Filippo Simini, Marta C. González, Amos Maritan, and Albert-László Barabási. “A Universal Model for Mobility and Migration Patterns”. In: *Nature* 484.7392 (2012), pp. 96–100 (cit. on p. 9).

[66] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. “Dropout: A Simple Way to Prevent Neural Networks from Overfitting”. In: *The Journal of Machine Learning Research* 15.1 (Jan. 1, 2014), pp. 1929–1958 (cit. on pp. 24, 26, 36).

[67] Niek Tax, Ilya Verenich, Marcello La Rosa, and Marlon Dumas. “Predictive Business Process Monitoring with LSTM Neural Networks”. In: 10253 (2017), pp. 477–492. arXiv: `1612.02130 [cs, stat]` (cit. on p. 25).

[68] Anna Triandafyllidou. *Handbook of Migration and Globalisation*. Edward Elgar Publishing, 2018 (cit. on p. 6).

[69] United Nations Department of Economic and Social Affairs UN DESA. *Recommendations on Statistics of International Migration: Revision 1*. New York: United Nations, 1998 (cit. on p. 5).

[70] United Nations Department of Economic and Social Affairs Population Division UN DESA. *International Migrant Stock 2019*. EN. `https://www.un.org/en/development/desa/population/index.asp`. United Nations Database. 2019 (cit. on p. 1).

[71] United Nations Department of Economic and Social Affairs Population Division UN DESA. *International Migration Flows to and from Selected Countries: The 2015 Revision*. EN. https://www.un.org/en/development/desa/population/migration/data/empirical2/migrationflows.asp. United Nations Database. 2015 (cit. on p. 6).

[72] Hal R. Varian. "Big Data: New Tricks for Econometrics". en. In: *Journal of Economic Perspectives* 28.2 (May 2014), pp. 3–28 (cit. on p. 7).

[73] Larry Wasserman. *All of Statistics: A Concise Course in Statistical Inference*. en. Springer Texts in Statistics. New York: Springer-Verlag, 2004 (cit. on p. 16).

[74] World Bank. *World Development Indicators*. 2020. URL: https://datacatalog.worldbank.org/dataset/world-development-indicators (cit. on p. 18).

[75] xkcd. *Machine Learning*. May 17, 2017. URL: https://xkcd.com/1838/ (visited on Mar. 10, 2020) (cit. on p. xiii).

[76] Kaisheng Yao, Trevor Cohn, Katerina Vylomova, Kevin Duh, and Chris Dyer. "Depth-Gated LSTM". In: (Aug. 25, 2015). arXiv: 1508.03790 [cs] (cit. on p. 25).

List of Keywords

A

Table A.1 contains the set of main keywords: *For GTI data retrieval, both singular and plural as well as male and female forms of these keywords are used where applicable. In the English language, both British and American English spelling is used. All French and Spanish keywords were included with and without accents* [9, Table 1].

Tab. A.1.: List of main keywords [9, Table 1].

English	French	Spanish
applicant	candidat	solicitante
arrival	arrivee	llegada
asylum	asile	asilo
benefit	allocation sociale	beneficio
border control	controle frontiere	control frontera
business	entreprise	negocio
citizenship	citoyennete	ciudadania
compensation	compensation	compensacion
consulate	consulat	consulado
contract	contrat	contrato
customs	douane	aduana
deportation	expulsion	deportacion
diaspora	diaspora	diaspora
discriminate	discriminer	discriminar
earning	revenu	ganancia
economy	economie	economia
embassy	ambassade	embajada
emigrant	emigre	emigrante
emigrate	emigrer	emigrar
emigration	emigration	emigracion
employer	employer	empleador
employment	emploi	empleo
foreigner	etranger	extranjero
GDP	PIB	PIB
hiring	embauche	contratacion

Tab. A.1.: (continued)

English	French	Spanish
illegal	illegal	ilegal
immigrant	immigre	inmigrante
immigrate	immigrer	inmigrar
immigration	immigration	inmigracion
income	revenu	ingreso
inflation	inflation	inflacion
internship	stage	pasantia
job	emploi	trabajo
labor	travail	mano de obra
layoff	licenciement	despido
legalization	regularisation	legalizacion
migrant	migrant	migrante
migrate	migrer	migrar
migration	migration	migracion
minimum	minimum	minimo
nationality	nationalite	nacionalidad
naturalization	naturalisation	naturalizacion
passport	passeport	pasaporte
payroll	paie	nomina
pension	retraite	pension
quota	quota	cuota
recession	recession	recesion
recruitment	recrutement	reclutamiento
refugee	refugie	refugiado
remuneration	remuneration	remuneracion
required documents	documents requis	documentos requisito
salary	salaire	sueldo
Schengen	Schengen	Schengen
smuggler	trafiquant	traficante
smuggling	trafic	contrabando
tax	tax	impuesto
tourist	touriste	turista
unauthorized	non autorisee	no autorizado
undocumented	sans papiers	indocumentado
unemployment	chomage	desempleo
union	syndicat	sindicato

Tab. A.1.: (continued)

English	French	Spanish
unskilled	non qualifies	no capacitado
vacancy	poste vacante	vacante
visa	visa	visa
waiver	exemption	exencion
wage	salaire	salario
welfare	aide sociale	asistencia social

Searching for a Better Life's Models Results

B

Figure B.1 and B.2 respectively present the results of the different estimations based on the unilateral 3.4 and bilateral 3.5 model [9].

Unilateral model including Google Trends Indices.

Sample	(1) Benchmark	(2) GTI	(3) Benchmark	(4) GTI	(5) Benchmark	(6) GTI	(7) Benchmark	(8) GTI
	All		Extended controls		Spoken Language > 50%		Internet Access > 10%	
Log GDP (origin)	−0.641***	−0.486**	−0.276	−0.340	−0.996***	−0.801***	−1.038***	−0.821**
	(0.231)	(0.191)	(0.245)	(0.233)	(0.283)	(0.211)	(0.390)	(0.314)
Log Population (origin)	2.161***	1.681***	0.690	0.729	1.730**	1.368*	2.154**	1.817**
	(0.597)	(0.624)	(0.880)	(0.847)	(0.788)	(0.719)	(0.859)	(0.759)
Unemployment rate			0.0225	0.00671				
			(0.0161)	(0.00959)				
Share of young population			0.0306	−0.00707				
			(0.0293)	(0.0307)				
State Fragility Index			0.000270	0.00534				
			(0.0109)	(0.0122)				
Polity IV Autocracy Score			−0.000771	−0.000536				
			(0.000950)	(0.000873)				
Mobile cellular subscriptions			−0.00191	−0.00117				
			(0.00160)	(0.00149)				
Internet users			−0.00903***	−0.00767***				
			(0.00265)	(0.00288)				
No. weather-related disasters			−0.00137	−0.00251				
			(0.00580)	(0.00654)				
No. non-weather-related disasters			−0.0189*	−0.0109				
			(0.0111)	(0.00915)				
GTI (unilateral)	–	✓	–	✓	–	✓	–	✓
Joint significance GTI (p-value)	–	0.000	–	0.000	–	0.000	–	0.000
Fixed effects								
Origin	✓	✓	✓	✓	✓	✓	✓	✓
Year	✓	✓	✓	✓	✓	✓	✓	✓
Observations	1068	1068	700	700	732	732	647	647
Number of origins	98	98	70	70	67	67	79	79
R^2 (within)	0.062	0.242	0.166	0.355	0.074	0.316	0.092	0.422
R^2 (overall)	0.988	0.991	0.987	0.990	0.990	0.992	0.992	0.995
Observations per predictor	534	15.5	107	9.1	366	10.6	324	9.4

Notes: Each column displays the result of a separate regression based on equation (1). Dependent variable is the logarithm of the annual migration flow (plus one) from a given origin country to all OECD destinations. Robust standard errors, clustered at the origin country level, in parentheses. *** $p < 0.01$, ** $p < 0.05$, * $p < 0.1$.
Sources: Authors' calculations based on OECD IMD 2004–2015, World Development Indicators, Google Trends, Polity IV, State Fragility Index, and EM-DAT International Disasters Database.

Fig. B.1.: Results of the unilateral model [9, Table 4]

Bilateral model including Google Trends Indices.

Specification	(1) Benchmark	(2) GTI	(3) Benchmark	(4) GTI	(5) Benchmark	(6) GTI	(7) Benchmark	(8) GTI	(9) Benchmark	(10) GTI
	(A)		(B)		(C)		(D)		(E)	
Log GDP (destination)	0.523***	0.154	0.494**	0.147			0.832***	0.791***		
	(0.203)	(0.222)	(0.207)	(0.230)			(0.185)	(0.182)		
Log Population (destination)	−1.845***	−0.0920	−1.802***	−0.0427			−1.911***	−1.899***		
	(0.364)	(0.433)	(0.357)	(0.424)			(0.325)	(0.324)		
Log GDP (origin)	−0.450***	−0.0814								
	(0.0989)	(0.117)								
Log Population (origin)	0.456*	1.299***								
	(0.276)	(0.320)								
GTI (bilateral)	–	✓	–	✓	–	✓	–	✓	–	✓
Joint significance GTI (p-value)	–	0.000	–	0.000	–	0.000	–	0.000	–	0.000
Fixed effects										
Destination	✓	✓	✓	✓	–	–	–	–	–	–
Origin	✓	✓	–	–	–	–	–	–	–	–
Year	✓	✓	–	–	–	–	–	–	–	–
Destination-year	–	–	–	–	✓	✓	–	–	✓	✓
Origin-year	–	–	✓	✓	✓	✓	✓	✓	✓	✓
Destination-origin	–	–	–	–	–	–	✓	✓	✓	✓
Observations	23,947	23,947	23,947	23,947	23,947	23,947	23,947	23,947	23,947	23,947
Number of pairs	2627	2627	2627	2627	2627	2627	2627	2627	2627	2627
R^2 (within)	0.001	0.272	0.001	0.299	0.000	0.311	0.007	0.0244	0.000	0.014
R^2 (overall)	0.732	0.805	0.735	0.814	0.739	0.820	0.971	0.971	0.974	0.974
Observations per predictor	5987	174	11,974	176	–	179	11,974	176	–	179

Notes: Each column displays the result of a separate regression based on equation (2). Dependent variable is the logarithm of the annual flow of migrants (plus one) from a given origin country to a specific OECD destination. Robust standard errors, clustered at the origin country level, in parentheses. *** $p < 0.01$, ** $p < 0.05$, * $p < 0.1$.
Sources: Authors' calculations based on OECD International Migration Database 2004–2015, World Development Indicators, and Google Trends.

Fig. B.2.: Results of the bilateral model [9, Table 5]

Data Extraction

C

The source code is available on the following repository:

`https://github.com/aia-uclouvain/gti-mig-paper`

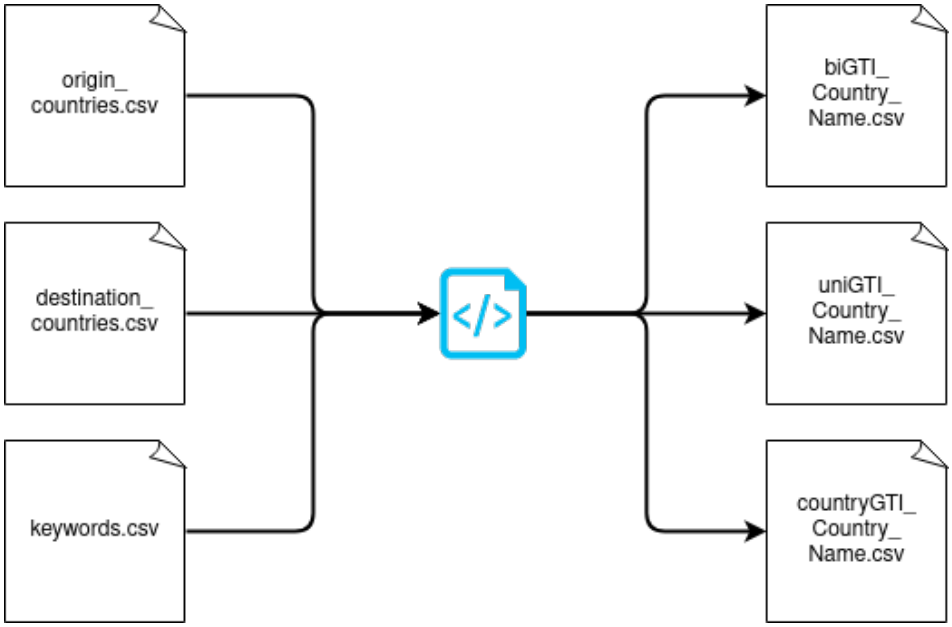

Fig. C.1.: Data extraction script from Google Trends using the pytrends API [25] – Input files are on the left side, while output files are on the right side.

The data originates from two sources:

- (i) *Google Trends (GT)* using the unofficial API *pytrends* [25];
- and (ii) the OECD's *International Migration Database (IMD)* specifically the data on *inflows of foreign population by nationality* [54].

Pytrends is an unofficial API for Google Trends. It "*allows a simple interface for automating downloading of reports from Google Trends*" and it's "*main feature is to allow the script to login to Google to enable a higher rate limit*" [25]. It is open source, under an Apache License 2.0. The rate limits are the following after some experimenting:

- 1600 sequential requests over a 12 hours time frame;
- 60 seconds of sleep between requests (successful or not) once you reach the limit.

The data from Google Trends spans from January 2004 to December 2015, while the IMD spans from 2004 to 2015. We do not take values before 2004 since GT only started in January 2004, while we do not take any value after 2015 from the IMD since there are no values available.

We use Google Trends Index (GTI) as proposed in Böhme et al. [9][1] to to build 3 different types of datasets:

- 1 bidirectional GTI: keyword and country name, e.g., "*visa Belgique*". See description for the `biGTI_Country_Name.csv` below;
- 2 unidirectional GTI: single keyword or country name, e.g., "*visa*" or "*Belgique*". See description for both the `uniGTI_Country_Name.csv` and the `countryGTI_Country_Name.csv` below.

To extract the data from GT we use a python script as shown in Figure C.1. The 3 types of input files are:

origin_countries.csv a list of English, French or Spanish speaking countries (101 countries), where the language is spoken by more than 50% of the population;

destination_countries.csv a list of the 36 OECD countries, where migration data is available or partially available;

keywords.csv a list of 67 keywords related to migration according to Böhme et al. [9].

For each of these 3 types there are 3 files by language: one in English; one in French; one in Spanish.

Once we extract the GT data, we transform it into a GTI. The 3 types of output files are the GTI spanning from January 2004 to January 2015 from one of the origin countries:

[1] "*the GTI data consists of high-frequency time series capturing the relative search intensities for any keyword performed through the Google search engine across the globe. The GTI is by far the most representative data source for online searches worldwide with Google having a market share of more than 80% on desktop devices. This figure increases to 97% once considering mobile and tablet device*" [9, p. 3]

biGTI_Country_Name.csv where the GT query is the combination of one of the keywords with one of the OECD countries; the file is structured by starting with a row containing the ISO code name of the destination country followed by 67 rows with 16 columns corresponding to each of the years from 2004 to 2015;

uniGTI_Country_Name.csv where the query is one of the OECD country's name; the file is structured as rows of 17 columns, the first column being the used keyword, and the 16 other columns corresponding to each of the years from 2004 to 2019;

countryGTI_Country_Name.csv where the GT query is one of the keywords. the file is structured as rows of 17 columns, the first column being the used country name, and the 16 other columns corresponding to each of the years from 2004 to 2015.

Notice that they are a lots of 0s in the extracted data. This will have to be taken into account while building our LSTM model.

ANN Approach: Validation of Hyper-parameters

D

Figures D.1, D.2, and D.3 present the results of the validation of the different hyper-parameters for the ANN approach.

Each figure displays the comparison of the different models for a specific variation of hyper-parameter. A figure is composed of 5 graphics, the first one showing the loss value and the others showing the values of 4 of the presented metrics (see 4.1) depending on the value of the tuned hyper-parameter and the number of epochs.

It is important to notice that these metrics are calculated on the standardised output values (ranged from 0 to 1).

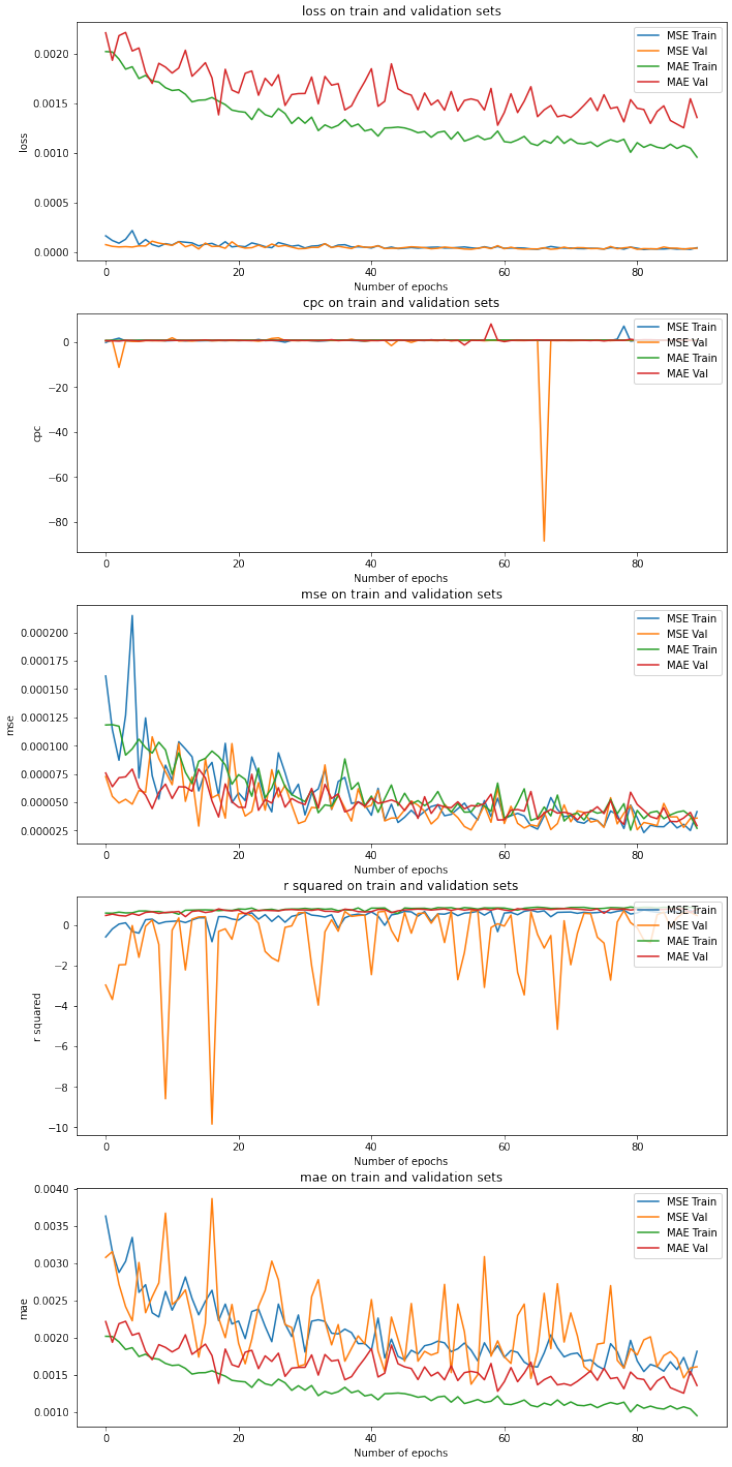

Fig. D.1.: Results of the validation for the loss function parameter for the ANN approach. The performance of the model using a CPC loss is not presented because the values obtained for some metrics were so poor it made the graph difficult to analyse.

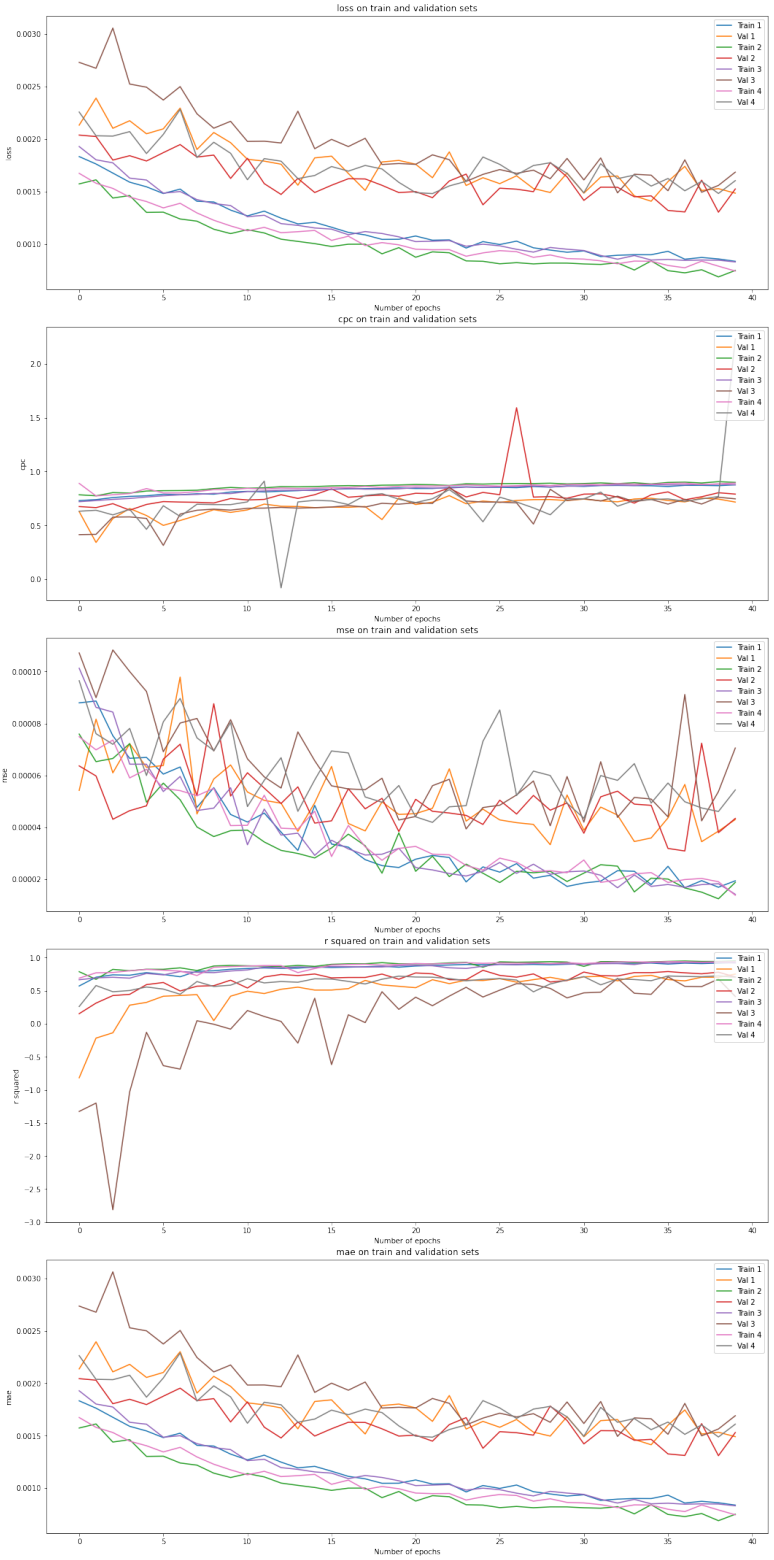

Fig. D.2.: Results of the validation for the number (depth) and the sizes (width) of hidden layers for the ANN approach. Model 1: 2 layers of width 50; model 2: 2 layers of size 200; model 3: 2 layers of size 100; and model 4: 3 layers of size 50.

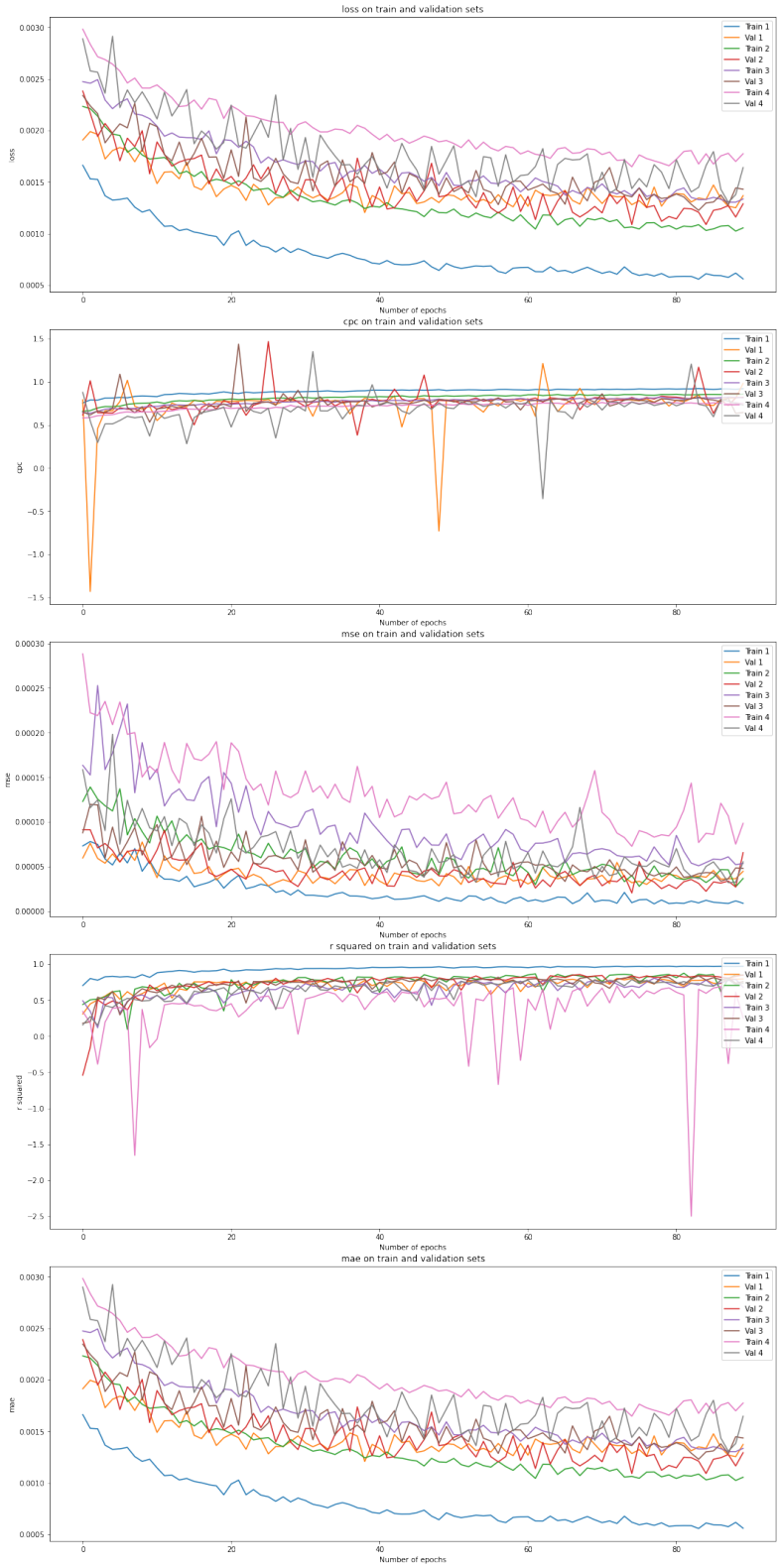

Fig. D.3.: Results of the validation for the dropout. Dropout of the different models: 1 - 0.0, 2 - 0.1, 3 - 0.2, 3 - 0.3.

LSTM Approach: Validation of Hyper-parameters

E

Figures E.1 and E.2 present the results of the validation of different hyper-parameters for the LSTM approach.

Each figure displays the comparison of the different models for a specific variation of hyper-parameter. A figure is composed of 5 graphics, each one showing the values of one of the presented metrics (see 4.1) depending on the value of the tuned hyper-parameter and the number of epochs.

Fig. E.1.: Results of the validation for the loss function parameter for the LSTM approach.

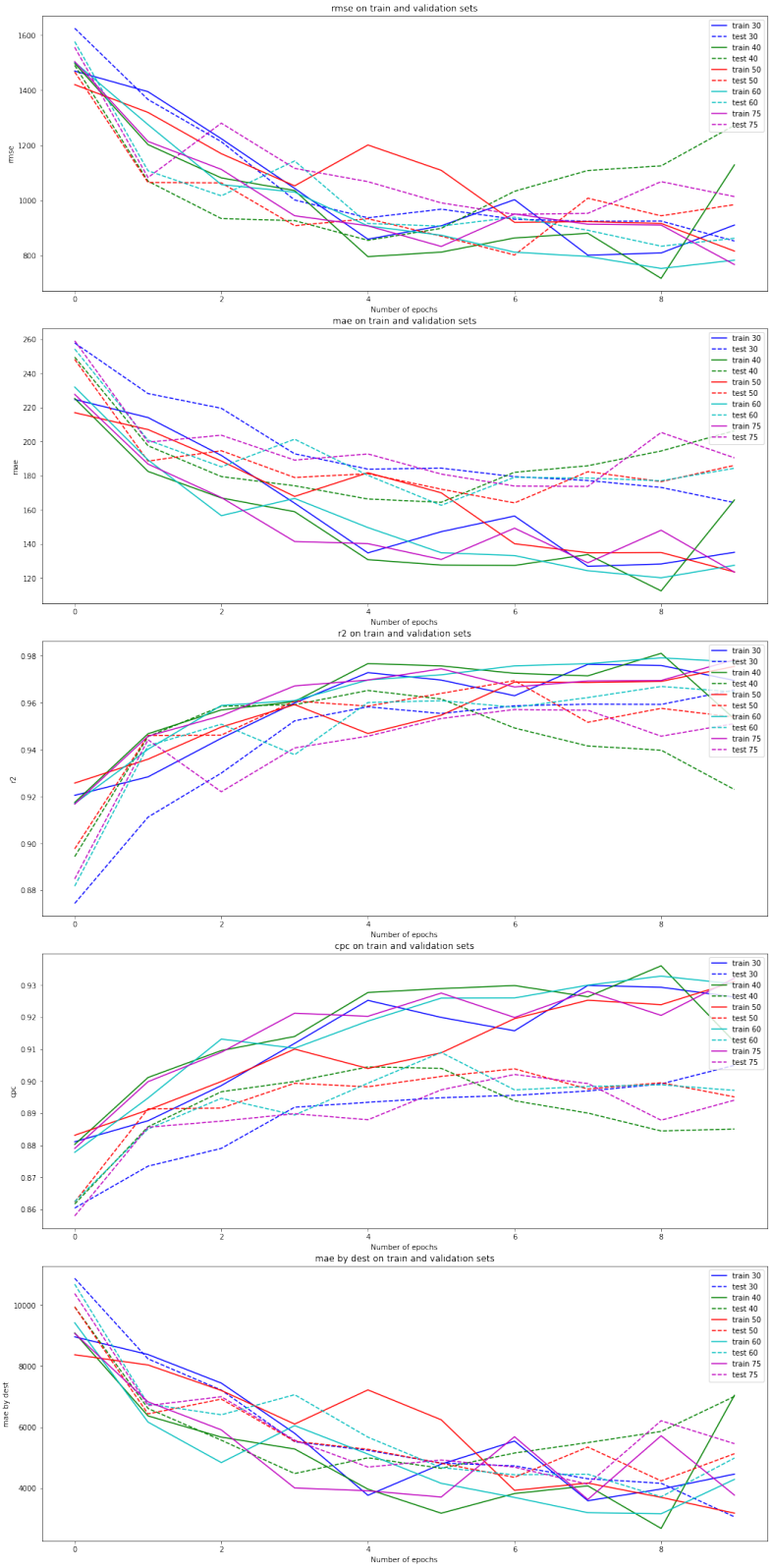

Fig. E.2.: Results of the validation of the size (depth) of the unique hidden layer for the LSTM approach. The loss function used by the different models is the CPC loss.

UNIVERSITÉ CATHOLIQUE DE LOUVAIN
École polytechnique de Louvain
Rue Archimède, 1 bte L6.11.01, 1348 Louvain-la-Neuve, Belgique | **www.uclouvain.be/epl**