Forecasting Performance of Regional Innovation Systems using Semantic-Based Genetic Programming with Local Search Optimizer

Petr Hajek[a], Roberto Henriques[b], Mauro Castelli[b*], Leonardo Vanneschi[b]

[a] Institute of System Engineering and Informatics, Faculty of Economics and Administration, University of Pardubice
Studentska 84, 532 10 Pardubice, Czech Republic
petr.hajek@upce.cz

[b] NOVA IMS Information Management School, Universidade Nova de Lisboa
1070-312, Lisboa, Portugal
{roberto,lvanneschi}@novaims.unl.pt

[*]corresponding author - mcastelli@novaims.unl.pt
NOVA IMS Information Management School 1070-312, Lisboa, Portugal
phone: +351 213 828 610

Abstract

Innovation performance of regional innovation systems can serve as an important tool for policymaking to identify best practices and provide aid to regions in need. Accurate forecasting of regional innovation performance plays a critical role in the implementation of policies intended to support innovation because it can be used to simulate the effects of actions and strategies. However, innovation is a complex and dynamic socio-economic phenomenon. Moreover, patterns in regional innovation structures are becoming increasingly diverse and non-linear. Therefore, to develop an accurate forecasting tool for this problem represents a challenge for optimization methods. The main aim of the paper is to develop a model based on a variant of genetic programming to address the regional innovation performance forecasting problem. Using the historical data related to regional knowledge base and competitiveness, the model should accurately and effectively predict a variety of innovation outputs, including patent counts, technological and non-technological innovation activity and economic effects of innovations. We show that the proposed model outperforms state-of-the-art machine learning methods.

1. Introduction

Innovation is considered as a complex and dynamic, socio-technical, socio-economic, socio-political phenomenon that has been recognised as a central issue in economic development [1] The concept of regional innovation systems has recently received increased attention [2,3] mainly due to the growing importance of regions (and other sub-national entities) in a globalised economy [4]. In regional innovation systems, private and public actors intensively interact and thus promote the generation, use, and dissemination of knowledge [5]. In addition, regions are critical entities for innovation policymaking because regions provide favourable conditions for knowledge creation and transfer [3]. In this context, measuring the innovation performance of regions has become a priority in order to develop integrated benchmarking systems in the knowledge-based economies [1]. This enables policymakers not only to comparatively evaluate the performance of regional innovation systems but also to identify best practices (innovation leaders) and target the regions in need (lagging behind regions). This is also why regional innovation performance is annually measured in many countries, for example using the innovation scoreboard for EU regions [6].

The advantages of using the indicators of innovation performance at regional level can be summarized as follows [7]: (1) analysts and statisticians have strong experience with the

collection and use of such indicators; (2) these indicators comprehensively cover all countries, industries and technological fields; (3) long time-series data are available to study the dynamics in the innovation performance of firms and industries across regions. In addition, the input-output innovation relationship is considered to be more robust at the regional level compared with the firm level [8]. This is attributed to both the important role of the regional context and the existence of externalities. Indeed, the results of firm-level models may provide incorrect inference in the presence of a strong effect of the regional context on the generation of innovations [9]. However, note that the results obtained at the regional level cannot be interpreted at the level of individual firms as these results might significantly differ from those obtained from firm-level data due to biased estimates [9].

The main concern in measuring regional innovation performance is the complexity and dynamic changes in regional innovation systems [2]. As a result, the data for the evaluation quickly become obsolete. Therefore, an accurate and reliable forecasting tool to support decision making presents a challenging task for optimization methods. Non-linear machine learning methods such as fuzzy rule-based systems and neural networks have been used for innovation forecasting at the firm level [10,11]. These methods outperformed traditional statistical forecasting models in terms of accuracy, indicating non-linear patterns in firm innovation activities. In addition, recent empirical evidence provides support for this assumption also at both the regional [12] and national level [13]. Moreover, chaos theory was used to detect non-linearity and strange attractors in the evolutionary path of patent counts [14]. Regarding innovation systems, Samara et. al. [15] developed an integrated system dynamics approach to analyse the impact of innovation policies on the performance of national innovation systems. However, no previous research known to us has forecasted the performance of regional innovation systems using artificial intelligence methods. The main advantage of these methods, compared with traditional statistical forecasting methods, is that no complex mathematical formulation of the input-output relationships is necessary. Moreover, traditional methods are not suitable for modelling phenomena characterised by a high variance [16]. In the case of regional innovation systems, the high variance is mainly due to the highly dynamic socio-economic environment. To address these issues, we develop a forecasting model based on genetic programming in this study. Specifically, we use a recently proposed and very promising variant of standard genetic programming that integrates the concept of semantic awareness and local search optimizers to generate forecasting models. We argue that this model is more appropriate to model intrinsic non-linear character of innovation performance than traditional statistical and machine learning forecasting models

because genetic programming: (1) has an excellent evolvability on training data [17] and (2) is able to generalize the solution also on testing data [18]. Our approach combines two recent advancements in genetic programming, this is (1) geometric semantic operator (GSO) that eliminates local optima by inducing a unimodal error surface on any forecasting algorithm and (2) local search optimizer (LSO) to make the convergence faster. The main idea of combining these approaches is to achieve a balance between exploration (GSO) and exploitation (LSO). As a result, the forecasting model can be optimized faster and overfitting can be avoided.

To verify the appropriateness of the proposed model for forecasting performance of regional innovation systems, the data on European regions for the period 2004-2012 were used. The inputs of the forecasting model are represented by indicators related to the regional knowledge base (regional knowledge generation, absorption, and transfer capacity) and regional competitiveness indexes approximating regional socio-technical, socio-economic and socio-political environment. The outputs include four indicators of the performance of regional innovation systems, namely patent counts, technological and non-technological innovation activity and economic effects of innovations. The models are first trained to forecast innovation performance for 2010, and then the models are tested on 2012 data. We demonstrate that the proposed model outperforms other statistical and artificial intelligence methods in terms of accuracy on testing data.

The remainder of this paper is structured as follows. In the next section, we present the inputs and outputs of the model and describe the data used. The variant of genetic programming proposed in this study to the forecasting problem is introduced in the following section. Section 4 describes the setting of the forecasting model and provides the experimental results comparing the proposed approach to other variants of genetic programming algorithm and other state-of-the-art forecasting methods. Finally, we conclude the paper, highlighting the main contributions of this study.

2. Data

Four interacting categories of determinants have been introduced into the models of regional innovation systems, namely regional competitiveness, knowledge generation, knowledge absorption and knowledge transfer [2,3,5,15]. Table 1 presents the determinants of regional innovation performance used in this study.

Different socio-economic conditions and regional competitiveness have been reported as an important determinant of regional innovation performance [19]. In fact, the specific combination of these conditions affects the ability of the region to transform R&D into innovation [20]. Business activity (measured as the share of self-employed persons) has been identified as a critical factor of innovation-based regional growth [21]. The index of labour market efficiency includes, among others, job mobility and labour productivity [22]. Technology readiness is of key importance for the overall competitiveness of a region. It can be characterised by ICT penetration rates and a regulatory framework which is friendly to ICT. Infrastructure index represents various dimensions of infrastructure, such as density, connectivity, and availability of roads, railways and flights.

Market size includes both the size of the regional market and the size of the potential market in neighbouring regions [22]. The institutional quality index measures the quality of political and regulatory framework.

R&D expenditure has been recognised as an important determinant of innovation performance in prior studies, indicating the level of knowledge generation within firms, research institutes and universities [23]. Different types of research are carried out by these organisations. The diffusion of technology is promoted by non-R&D innovation expenditures. The generation of scientific (analytical) and engineering (synthetic) knowledge is approximated by human resources in science and high-tech sectors [24,25].

Indicators of education estimate regional capacity to absorb knowledge and technology. The level of advanced academic skills can be measured by the share of the population with a tertiary education, while the participation of people in life-long learning represents readiness to continue technological development and innovation [26].

In regional innovation systems, intensive economic and knowledge interactions take place, both within firms as well as between firms and universities or research centres. In other words, the borders of the system are given by the intensity of these interactions. Inter-regional knowledge transfer was estimated by spatial centrality and employment in neighbouring regions. Spatial centrality index measures the remoteness (time accessibility) of the regions [27]. More frequent interactions promote knowledge transfer between regions. Reportedly, the performance of a regional innovation system strongly depends on the performance in neighbouring regions [28]. Joint publications between firms and universities approximate the knowledge transfer between researchers from the private and public sectors. A wide range of collaborating actors and knowledge sources helps firms achieve and sustain innovation [29].

This cooperation leads to the transfer of knowledge between firms. Therefore, the share of collaborating firms was used as the indicator of knowledge transfer in this study.

Table 1 - Input and output attributes for forecasting performance of regional innovation systems

	Input attributes – regional competitiveness	*Source of data*
x_1	*Business activity*	*Eurostat*
x_2	*Labour market efficiency*	*RCI*
x_3	*Technology readiness*	*RCI*
x_4	*Infrastructure index*	*RCI*
x_5	*Market size*	*RCI*
x_6	*Institutional quality*	*RCI*
	Input attributes – knowledge generation	
x_7	*R&D expenditure – private*	*Eurostat*
x_8	*R&D expenditure – government*	*Eurostat*
x_9	*R&D expenditure – higher education*	*Eurostat*
x_{10}	*Non-R&D innovation expenditure*	*Eurostat*
x_{11}	*Human resources in science*	*Eurostat*
x_{12}	*Employment in high-tech sectors*	*Eurostat*
	Input attributes – knowledge absorption	
x_{13}	*Tertiary education*	*Eurostat*
x_{14}	*Lifelong education*	*Eurostat*
	Input attributes – knowledge transfer	
x_{15}	*Spatial centrality*	[27]
x_{16}	*Inter-regional employment*	*Eurostat*
x_{17}	*Joint scientific publications (firms and universities)*	*Eurostat*
x_{18}	*SME collaboration*	*Eurostat*
	Output attributes – innovation performance	
y_1	*Patent counts (applications to the EPO)*	*Eurostat*
y_2	*Technological innovation*	*Eurostat*
y_3	*Nontechnological innovation*	*Eurostat*
y_4	*Sales of new products*	*Eurostat*

Patent counts not only measure the level of technological development but they also represent a proxy of the regional innovation capacity and evaluate the productivity of investments in R&D [2]. However, the number of patented inventions is not a perfect measure of innovative activity because it over-emphasizes the effects of localised interactions [30]. Therefore, it is

necessary to introduce alternative output indicators of regional innovation activity, such as those included in the Frascati Manual of the OECD [31]. Adopting this approach, we estimated the performance of regional innovation systems by the proportion of technological (product/process) and nontechnological (marketing/organisational) innovators. Thus, the diversity of innovation processes was considered [32]. Finally, the economic success of innovation activity was measured by the sales of new products (new-to-market and new-to-firm products).

European regional innovation systems at the level of NUTS 2 (Nomenclature of Territorial Units for Statistics) regions was examined due to its increasing importance regarding innovation and R&D policies. Compared to national innovation systems, these regions are considered more homogenous [33]. Data were collected for 259 European regions for the period of 2004-2012 (the list of the regions is presented as Supplementary Material). Since the effects of the inputs are expected to be delayed, a two-year forecasting horizon was applied in agreement with previous studies [23]. Most data from Eurostat were collected from the Community Innovation Surveys which has two-year reference periods. Therefore, a two-year collection period was available.

3. An Introduction to Genetic Programming

Genetic Programming (GP) [34] is a computational method that belongs to the computational intelligence research area called evolutionary computation [35]. GP consists of the automated learning of computer programs by means of a process inspired by the theory of biological evolution of Darwin. In the context of GP, the word program can be interpreted in general terms, and thus GP can be applied to the particular cases of learning expressions, functions and, as in this work, data driven predictive models. In GP, programs are typically encoded by defining a set F of primitive functional operators and a set T of terminal symbols. Typical examples of primitive functional operators may include arithmetic operations (+, -, *, etc.), other mathematical functions (such as sin, cos, log, exp), or, according to the context and type of problem, also boolean operations (such as AND, OR, NOT), or more complex constructs such as conditional operations (such as If-Then-Else), iterative operations (such as While-Do) and other domain-specific functions that may be defined. Each terminal is typically either a variable or a constant, defined on the problem domain. The objective of GP is to navigate the space of all possible programs that can be constructed by composing symbols in F and T , looking for the most appropriate ones for solving the problem at hand. Generation by

generation, GP stochastically transforms populations of programs into new, hopefully improved, populations of programs. The appropriateness of a solution in solving the problem (i.e. its quality) is expressed by using an objective function (the fitness function). In synthesis, the GP paradigm breeds computer programs to solve problems by executing the following steps:

1. Generate an initial population of computer programs (or individuals).
2. Iteratively perform the following steps until the termination criterion has been satisfied:
    (a) Execute each program in the population and assign it a fitness value according to how well it solves the problem.
    (b) Create a new population by applying the following operations:
        i. Probabilistically select a set of computer programs to be reproduced, on the basis of their fitness (selection).
        ii. Copy some of the selected individuals, without modifying them, into the new population (reproduction).
        iii. Create new computer programs by genetically recombining randomly chosen parts of two selected individuals (crossover).
        iv. Create new computer programs by substituting randomly chosen parts of some selected individuals with new randomly generated ones (mutation).
3. The best computer program to appear in a generation is designated as the result of the GP process at that generation. This result may be a solution (or an approximate solution) to the problem.

The standard genetic operators [34] act on the structure of the programs that represent the candidate solutions. In other terms, standard genetic operators act at a syntactic level. More specifically, standard crossover is traditionally used to combine the genetic material of two parents by swapping a part of one parent with a part of the other.

Considering the standard tree-based representation of programs often used by GP [34], after selecting two individuals based on their tness, standard crossover chooses a random subtree in each parent and swaps the chosen subtrees between the two parents, thus generating new programs, the offspring. On the other hand, standard mutation introduces random changes in the structures of the programs in the population. For instance, the traditional and mostly used

mutation operator, called sub-tree mutation, works by randomly selecting a point in a tree, removing whatever is currently at the selected point and whatever is below the selected point and inserting a randomly generated tree at that point. As we clarify in Section 3, Geometric Semantic GP (GSGP) [17,36] uses genetic operators that are different from the standard ones, since they are able to act at the semantic level. The reader who is interested in a deeper discussion of GP is referred to [37] .

3.1. Symbolic Regression with Genetic Programming.

The problem tackled in this paper can typically be modelled as a symbolic regression problem. So, it is appropriate to introduce here the general idea of symbolic regression and the way in which this kind of problem is typically approached with GP. In symbolic regression, the goal is to search for the symbolic expression $T^O : R^p \rightarrow R$ that best fits a particular training set $TR = \{(\mathbf{x}_1, t_1), (\mathbf{x}_2, t_2), ..., (\mathbf{x}_n, t_n)\}$ of n input/outout pairs, with $\mathbf{x}_i \in R^p$ and $t_i \in R$. The general symbolic regression problem can then be defined as:

$$T^O \leftarrow argmin_{(TR \in G)} \; f(T(\mathbf{x}_i), t_i) \quad with \quad i = 1, 2, ..., n \tag{1}$$

Where G is the solution space defined by the primitive set (functions and terminals) and f is the fitness function, based on a distance (or error) between a program's output $T(\mathbf{x}_i)$ and the expected output, or target, t_i. In other words, the objective of symbolic regression is to find a function T^O (called data model) that perfectly matches the given input data into the known targets. In symbolic regression, the primitive set is generally composed of a set of functional symbols F containing mathematical functions (such as, for instance, arithmetic functions, trigonometric functions, exponentials, logarithms, etc.) and by a set of terminal symbols T containing p variables (one variable for each feature in the dataset), plus, optionally, a set of numeric constants.

4. Geometric Semantic Genetic Programming

Even though the term semantics can have several different interpretations, it is a common trend

in the GP community (and this is what we do also here) to define the semantics of a solution as the vector $\mathbf{s}(T) = [T(\mathbf{x}_1), T(\mathbf{x}_2), ..., T(\mathbf{x}_n)]$ of its output values. From this perspective, a GP individual can be identified by a point (its semantics $\mathbf{s}(T)$) in a multidimensional space that we

call *semantic space* (where the number of dimensions is equal to the number of observations in the training set (or training cases). The term Geometric Semantic Genetic Programming (GSGP)[17] indicates a recently introduced variant of GP in which traditional crossover and mutation are replaced by so-called Geometric Semantic Operators (GSOs), which exploit semantic awareness and induce precise geometric properties on the semantic space. GSOs, introduced by Moraglio et al. [36], are becoming more and more popular in the GP community [38] because of their property of inducing a unimodal error surface (i.e. an error surface characterized by the absence of locally optimal solutions on training data) on any problem consisting of matching sets of input data into known targets (like for instance supervised learning problems such as symbolic regression and classification). The interested reader is referred to [17] for an introduction to GSGP, where the property of unimodality of the error surface is carefully explained. Here, we report the definition of the GSOs as given by Moraglio et al. for real functions, since these are the operators we will use in this work. For applications that consider other types of data, the reader is referred to [36].

Geometric Semantic Crossover (GSXO). Given two parent functions T_1, T_2: $\mathrm{R}^n \rightarrow \mathrm{R}$, the geometric semantic crossover returns the real function $T_{XO} = (T_1 \cdot T_R) + ((1 - T_R) \cdot T_2)$, where T_R is a random function such that T_R: $\mathrm{R}^n \rightarrow [0; 1]$.

To constrain T_R in producing values in [0; 1] we use the sigmoid function $T_R = \frac{1}{1+e^{-Trand}}$ where $\mathrm{T_{rand}}$ is a random tree with no constraints on the output values.

Geometric Semantic Mutation (GSM). Given a parent function T: $\mathrm{R}^n \rightarrow \mathrm{R}$, the geometric semantic mutation with mutation step *ms* returns the real function $T_M = \mathrm{T} + ms \cdot (T_{R1} - T_{R2})$, where T_{R1} and T_{R2} are random real functions.

Moraglio and co-authors show that GSXO corresponds to geometric crossover in the semantic space (i.e. the point representing the offspring lies on the segment joining the points representing the parents) and GSM corresponds to box mutation on the semantic space (i.e. the point representing the offspring lies within a box of radius *ms*, centered in the point representing the parent). As Moraglio and co-authors point out, GSGP has an important drawback: GSOs create much larger offspring than their parents and the fast growth of the individuals in the population rapidly makes fitness evaluation unbearably slow, making the

system unusable. In [39], a possible workaround to this problem was proposed, consisting in an implementation of Moraglio's operators that makes them not only usable in practice, but also very efficient. With this implementation, the size of the evolved individuals is still very large, but they are represented in a particularly clever way (using memory pointers and avoiding repetitions) that allows us to store them in memory efficiently. So, using this implementation, we are able to generate very accurate predictive models, but these models are very large.

5. Local Search in GP and GSGP

In Section 5.1, we discuss previous approaches for integrating Local Search (LS) with standard GP. Afterwards, in Section 5.2, we present the first integration of a local searcher within GSGP.

5.1. Local Search in Standard GP

Many works have appeared so far on how to combine an evolutionary algorithm with a local optimizer. In general, such approaches are considered to be a simple type of memetic search [40]. The basic idea is straightforward: include within the optimization process an additional search operator that takes an individual (or several) as an initial point and searches for the local optima around it. Such a strategy can help ensure that the local region around each individual is fully exploited. However, there can be some negative consequences to such an approach. The most evident is the computational overhead: while the cost of a single LS might be negligible, performing it on every individual might become inefficient. Second, LS can produce overfitted solutions, stagnating the search on local optima. These issues aside, these techniques have produced impressive results in a variety of scenarios, some of which are reviewed by Chen et al. [41]. A noteworthy aspect of this survey is an almost complete lack of papers that deal with GP. Of the more than two hundred papers covered by Chen et al., in fact, only a couple deal with memetic GP. This indicates that the GP community may have not addressed the topic adequately.

In [42], gradient descent is used to optimize numerical constants within a GP tree, achieving good results on five symbolic regression problems. Similarly, in [43] and [44] a LS algorithm is used to optimize the value of constant terminal elements. In [43] gradient descent is used and tested on classification problems, while [44] uses resilient backpropagation and evaluates the proposal on a real-world problem, in both cases leading towards improved results. In [45],

the authors include weight parameters for each function node, which the authors call inclusion factors; these weights modulate the importance that each node has within the tree. The authors also propose a series of new search operators that explicitly consider the parametrization of each GP tree. In a recent work [46,47], this problem was addressed by implementing a very simple parametrization of the tree, by constraining the number of internal parameters of each tree regardless of its size. Several different strategies were compared to determine when a local optimizer should be applied, showing that it is often best to apply it on either all the population or a subset of the best individuals. The LS used is called trust region optimization [48], and results showed substantial improvements in performance compared with standard GP search on several benchmark and real-world problems. A similar approach was developed by [46], with two noteworthy differences. First, parameters replace all constants present on a given tree, and each GP tree is enhanced by adding an artificial root tree that effectively adds a weight coefficient and a bias to the entire tree, then the Levenberg-Marquardt optimizer is used to find the optimal values for these parameters. Second, the authors apply constant optimization to the population using different probabilities, as well as a strict offspring selection variant for comparison.

5.2. Local Search in Geometric Semantic Genetic Programming

One of the systems studied in this work integrates a LS strategy within GSGP. In particular, a local searcher is included within the GSM operator, since previous works have shown that GSGP achieves its best performance using only mutation during the search [49]. In particular, the GSM with LS (GSM-LS) of a tree T generates an individual:

$$T_M = \alpha_0 + \alpha_1 \cdot T + \alpha_2 \cdot (T_{R1} - T_{R2})$$

where $\alpha_i \in \mathrm{R}$ and α_2 replaces the mutation step parameter *ms* used in GSM. This in fact defines a basic multi-variate linear regression problem, which could be solved, for example, by Ordinary Least Square regression (OLS). However, in this case we have n linear equations, where n is the number of fitness cases, and only three unknowns (the α_is). This gives an over-determined multivariate linear fitting problem, which can be solved through Singular-Value Decomposition (SVD) (in this work, the GNU Scientific Library available at http://www.gnu.org/software/gsl/ is used). We argue that this should be seen as a LS operator, that attempts to determine the best linear combination of the parent tree and the random trees used to perturb it, which is local in the sense of the linear problem posed by the GSM

operator. It should not be seen as a LS in the entire semantic space, since in that case the LS would necessarily converge to the optimum in this unimodal landscape.

This approach is similar to two previously proposed approaches. First, the linear fitting problem is reminiscent of the linear scaling procedure proposed in [50], which allows GP to fit the form of the desired output without necessarily optimizing the scale or bias. However, in that case, the scaling process is only used to adjust the fitness value of each individual, while the search operators used are standard ones. Second, and more closely related to this work, the non-isotropic Gaussian mutation proposed in [51], is used to perform a run-time analysis of GSGP. However, the mutation proposed in that work considers a fixed set of basis functions instead of randomly generated GP trees, and perturbs the linear combination with Gaussian-noise instead of providing the best fit coefficients. Finally, the work presented by [52] also uses a multivariate linear regression approach to optimize evolved solutions, with several key differences. Particularly, the search is conducted by standard GP, not GSGP, and each tree is decomposed into a set of subtrees which are then linearly combined. The method is much more explorative then the one presented here.

Moreover, the approach we propose contrasts with previous work [47], that relied on a non-linear local optimizer, since the linear assumption is mostly not satisfied by the expression evolved with standard GP and the corresponding parametrization. Instead, in this new approach, it is simple to apply an optimizer based on a linear regression, given that the GSM operator defines a linear expression in parameter space.

The idea of including a LS method is based on a very simple observation related to the properties of the geometric semantic operators: while these operators are effective in achieving good performance with respect to standard syntax-based operators, they require many generations to converge to optimal solutions. By including a local search method, we expect to improve the convergence speed of the search algorithm and to obtain better performance with respect to the algorithm that only uses GSOs. Moreover, by speeding up the search process, it will be possible to limit the construction of over-specialized solutions that, in the end, would overfit the data.

6. Experiments

This section describes the data pre-processing, experimental settings and the obtained results.

6.1. Data pre-processing

To pre-process the data, we first used ε-SVR (support vector regression with radial basis kernel function and penalty parameter C=8) to impute the missing values. For each attribute, all attributes except the missing one were used to estimate the missing value. A normalisation step into the range [0, 1] was applied to the input/output data using the *mapminmax* function. Basic descriptive statistics on the data are presented in Appendix 1.

6.2. Experimental settings

Input variables from the years 2004-2008 and output variables from 2010 were used as training data, while inputs from 2006-2010 and outputs from 2012 served as testing data, respectively. In addition to the determinants of innovation performance (x_1, x_2, …, x_{18}), the time-series component of the outputs was considered using the historical values of four performance indicators (y_1, y_2, y_3, y_4), this is from 2004-2008 to predict 2010 values in training data and from 2006-2010 to predict 2012 values in testing data, respectively (as presented in Figure 1). In other words, each of the 259 regions represented one sample in both the training and testing data and sliding window technique was used to perform time-series prediction. The number of input attributes in the data sets was 66 ((18 determinants of innovation performance + 4 performance indicators) × 3 time periods), see Figure 1 for details.

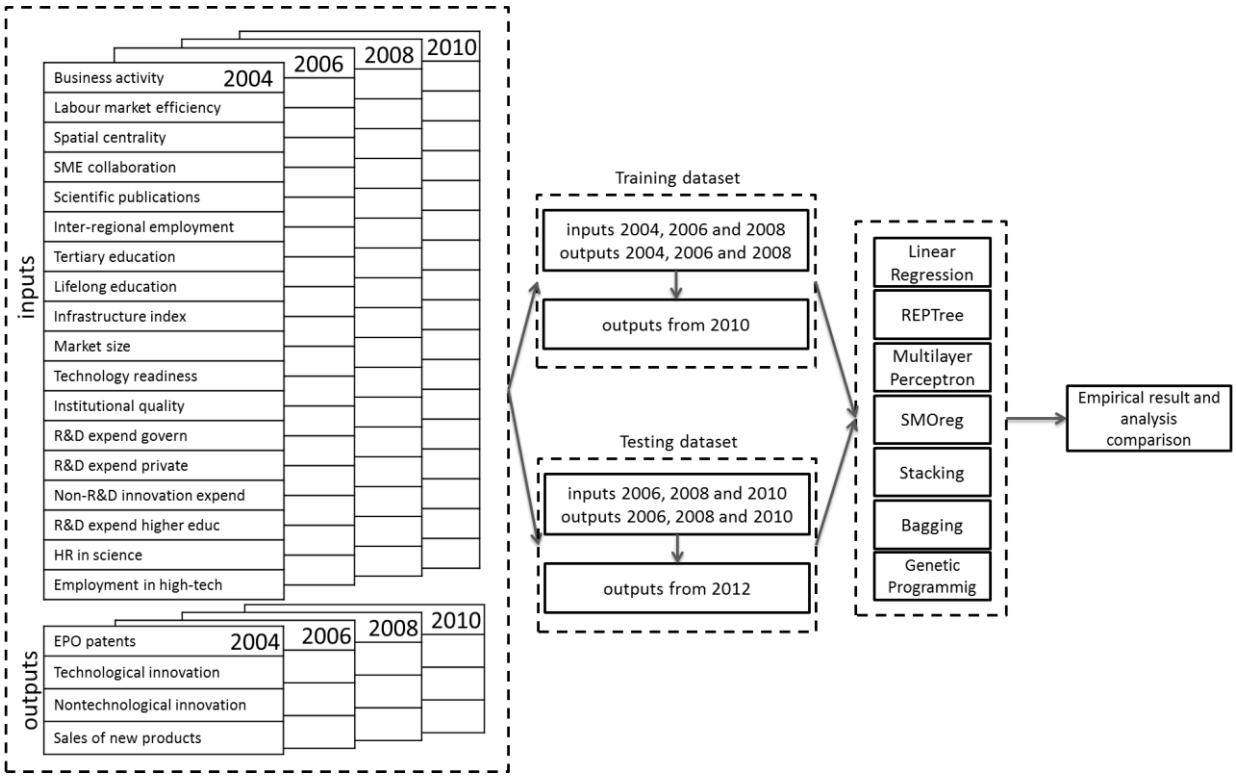


Figure 1 - Experimental settings. Time-series component of the outputs was considered using four performance indicators (outputs - EPO patents, Technological innovation, Nontechnological

innovation, and Sales of new products), from 2004-2008 to predict 2010 values (in training) from 2006-2010 to predict 2012 values (in testing).

As most real-world time series are nonlinear and/or stochastic, differencing data first is considered the best approach to developing an accurate artificial intelligence forecasting model [53]. Moreover, stationarity may improve forecasting performance on testing data regardless of the underlying data generating process. As we found clear evidence of non-stationarity for all four outputs (using Hadri Lagrange multiplier panel unit root test), we applied first differences to the data to reduce the effect of non-stationarity.

In order to validate our results, two different experiments were built to forecast the regional innovation performance. The first experiment (from now on, *experiment 1*) considers the original input and output values while in the second experiment (from now on, *experiment 2*) data differences to reduce the effect of non-stationarity were considered. The two different configurations are presented in Table 2 and Table 3.

Table 2 - Input and output variables used in Experiment 1. *Inp* stands for input variable and *Out* stands for output variable.

		x_1	x_2	x_3	x_4	x_5	…	x_{18}	y_1	y_2	y_3	y_4
Training Dataset	*2004*	*Inp*	*Inp*	*Inp*	*Inp*	*Inp*	*Inp*	*Inp*	*Inp*	*Inp*	*Inp*	*Inp*
	2006	*Inp*	*Inp*	*Inp*	*Inp*	*Inp*	*Inp*	*Inp*	*Inp*	*Inp*	*Inp*	*Inp*
	2008	*Inp*	*Inp*	*Inp*	*Inp*	*Inp*	*Inp*	*Inp*	*Inp*	*Inp*	*Inp*	*Inp*
	2010								*Out*	*Out*	*Out*	*Out*
	2012											
Testing Dataset	*2004*											
	2006	*Inp*	*Inp*	*Inp*	*Inp*	*Inp*	*Inp*	*Inp*	*Inp*	*Inp*	*Inp*	*Inp*
	2008	*Inp*	*Inp*	*Inp*	*Inp*	*Inp*	*Inp*	*Inp*	*Inp*	*Inp*	*Inp*	*Inp*
	2010	*Inp*	*Inp*	*Inp*	*Inp*	*Inp*	*Inp*	*Inp*	*Inp*	*Inp*	*Inp*	*Inp*
	2012								*Out*	*Out*	*Out*	*Out*

Table 3 - Input and output variables used in Experiment 2. *Inp* stands for input variable and *Out* stands for output variable.

		x_1	x_2	x_3	x_4	x_5	…	x_{18}	y_1	y_2	y_3	y_4
Training dataset	*2004_2006*	*Inp*	*Inp*	*Inp*	*Inp*	*Inp*	*Inp*	*Inp*	*Inp*	*Inp*	*Inp*	*Inp*
	2006_2008	*Inp*	*Inp*	*Inp*	*Inp*	*Inp*	*Inp*	*Inp*	*Inp*	*Inp*	*Inp*	*Inp*

	2008_2010								*Out*	*Out*	*Out*	*Out*
	2010_2012											
	2004_2006											
Testing	*2006_2008*	*Inp*	*Inp*	*Inp*	*Inp*	*Inp*	*Inp*	*Inp*	*Inp*	*Inp*	*Inp*	*Inp*
dataset	*2008_2010*	*Inp*	*Inp*	*Inp*	*Inp*	*Inp*	*Inp*	*Inp*	*Inp*	*Inp*	*Inp*	*Inp*
	2010_2012								*Out*	*Out*	*Out*	*Out*

As fitness, we used the Mean Absolute Value (MAE) between the output of the several algorithms and the corresponding target (i.e. expected output).

With respect to the GP systems, the following parameters were used: 50 runs were executed for each GP system. In all the experiments, the population size consists of 200 individuals and each run was left to evolve for 500 generations. The crossover probability was equal to 0.8, while the probability of mutation was 0.2, with a random mutation step as suggested in [49]. The individuals are initialized using the ramped half-and-half method [34], with the maximal initial depth equal to 6 and tournament selection [34] of size 6 was used. The functional operators were +, ×, −, and the protected division [34]. The set of terminals nodes consists of the input variables. For each run and for each generation, we recorded the fitness of the best individual in the population in the training set and the fitness of the same individual in the test set. The GSM-LS operator was used only in the first 20 generations to avoid overfitting, while in the remaining generation the standard geometric semantic mutation was considered.

6.3. Results

This section discusses the experimental results. The presentation is organised as follows: initially, a comparison between three different genetic programming systems is considered. In a second step, the best performer of the previous phase is compared against various machine learning techniques that are commonly used to address the problem at hand. In particular, the three systems taken into account in the first part are a standard syntax-based GP (ST-GP), a system that uses the geometric semantic genetic operators defined in Section 3 (GSGP) and a system that uses the proposed GSM-LS operator outlined in Section 3 (GSGP-LS). Results produced by the three systems are summarised, for all the problem configurations, in Figure 2 and Figure 3. As it is possible to note, including semantic awareness in the search process is usually beneficial, resulting in better performance (i.e., lower error) with respect to ST-GP.

Among GSGP and GSGP-LS the results are comparable, but it is important to remark that GSGP-LS is able to obtain the same good performance of GSGP in a reduced number of

generations. This fastest convergence is an important advantage, especially when a vast amount of data must be processed and analysed.

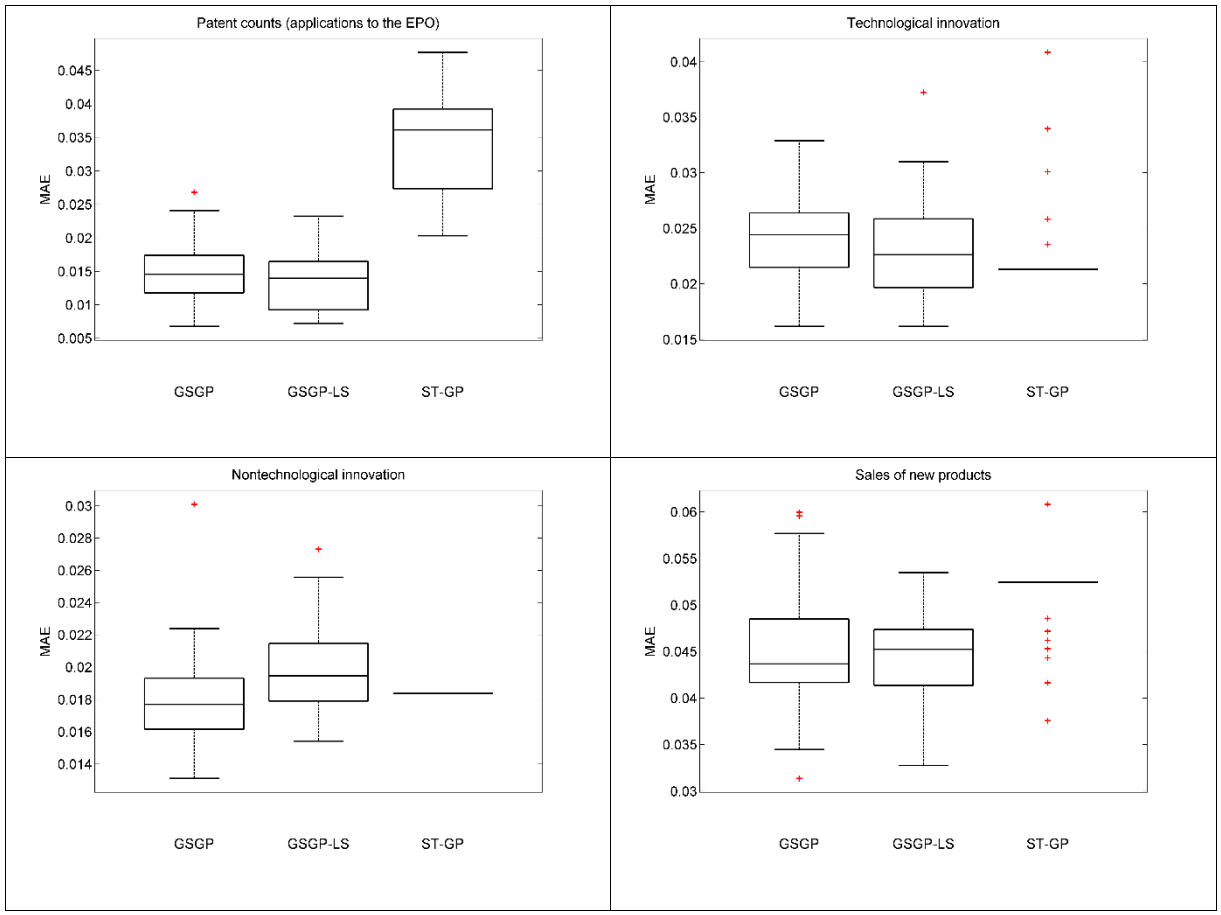
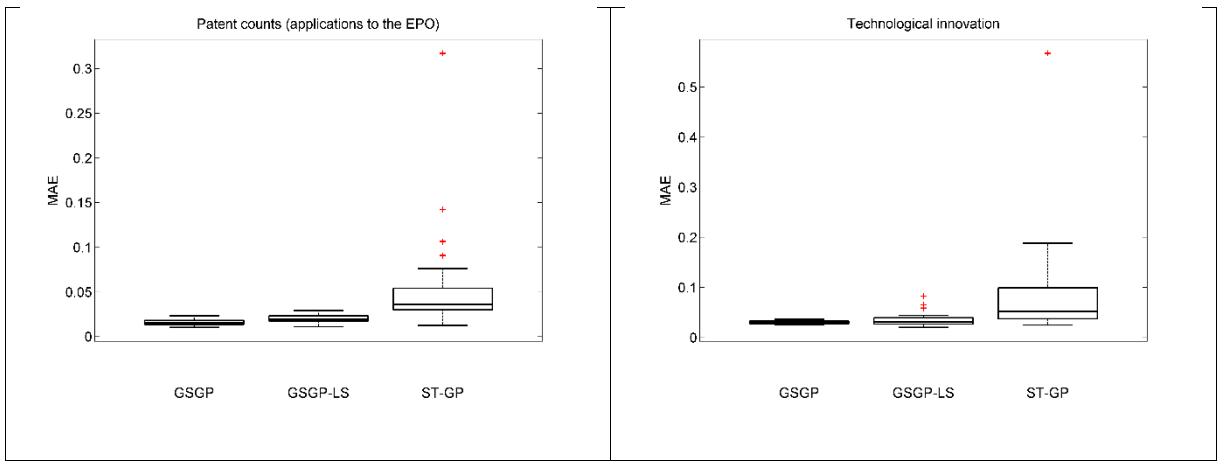


Figure 2 - Boxplots of mean absolute error (test fitness) for the 50 runs of the considered GP systems on *Experiment 1*. On each box, the central mark is the median, the edges of the box are the 25th and 75th percentiles, and the whiskers extend to the most extreme data points not considered outliers.



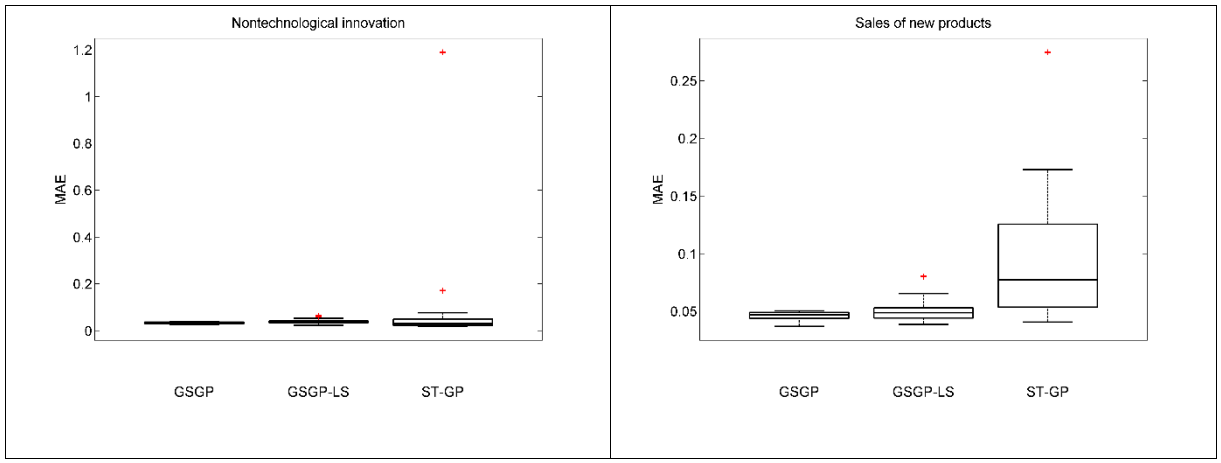


Figure 3 - Boxplots of mean absolute error (test fitness) for the 50 runs of the considered GP systems on *Experiment 2*. On each box, the central mark is the median, the edges of the box are the 25th and 75th percentiles, and the whiskers extend to the most extreme data points not considered outliers.

6.4. Comparison with Other Techniques

Besides comparing GSGP-LS against GSGP and ST-GP, it is also important to compare the performance of GSGP-LS against other well-known methods. In particular, we take into account linear regression (LR) [54], Multi-layer perceptron (MLP) [55] , Fast decision tree learner (REPTree) [56], support vector regression (SMOreg) [57], Stacking [58] and Bagging [59].

This comparison allows us to draw some considerations about the competitiveness of the results returned by GSGP-LS. To perform this experimental part, the implementations provided by the Weka public domain software [60] was considered. The values of the parameters characterizing the different techniques were chosen by performing a preliminary tuning phase. We used the functions provided by WEKA for finding the best parameter settings for the methods taken into account. In particular, the tuning phase has been performed by using the WEKA meta-classifier (CVParameterSelection). The meta-classifier provides a way of automating the tuning process.

Table 4 reports the median values of the training and test errors (MAE) of the solutions obtained by all the studied techniques. As it is possible to note, the GP-based methods are generally the best performers. Interestingly, MLP produces good results with respect to the other techniques on the training set, but it is not able to produce the same performance on the test set that is, ultimately, what matters most.

Results achieved by the different techniques on the test set, for the problem configurations considered, are reported in Figure 4 and Figure 5. From these box plots it is possible to draw similar conclusions with respect to the ones related to the first part of the experimental phase:

also when compared against state-of-the-art machine learning techniques, GSGP-LS is able to produce the best results.

Table 3 - Experimental comparison between different non-evolutionary techniques and GSGP-LS. Median of the training error and test error (MAE) calculated over 50 independent runs.

		Experiment 1		*Experiment 2*	
Model	*Output*	*Train*	*Test*	*Train*	*Test*
Bagging	y_1	*0.013*	*0.102*	*0.018*	*0.148*
	y_2	*0.038*	*0.097*	*0.047*	*0.124*
	y_3	*0.030*	*0.091*	*0.034*	*0.102*
	y_4	*0.041*	*0.198*	*0.053*	*0.223*
Linear Regression	y_1	*0.019*	*0.092*	*0.023*	*0.100*
	y_2	*0.054*	*0.158*	*0.074*	*0.140*
	y_3	*0.043*	*0.130*	*0.059*	*0.131*
	y_4	*0.052*	*0.251*	*0.074*	*0.183*
REPTree	y_1	*0.018*	*0.093*	*0.025*	*0.118*
	y_2	*0.043*	*0.127*	*0.053*	*0.135*
	y_3	*0.037*	*0.106*	*0.039*	*0.109*
	y_4	*0.055*	*0.199*	*0.066*	*0.233*
SMOreg	y_1	*0.017*	*0.099*	*0.021*	*0.115*
	y_2	*0.052*	*0.138*	*0.067*	*0.201*
	y_3	*0.040*	*0.123*	*0.054*	*0.134*
	y_4	*0.046*	*0.232*	*0.065*	*0.215*
Stacking	y_1	*0.025*	*0.097*	*0.025*	*0.173*
	y_2	*0.056*	*0.136*	*0.055*	*0.191*
	y_3	*0.035*	*0.113*	*0.042*	*0.125*
	y_4	*0.048*	*0.193*	*0.065*	*0.202*
MLP	y_1	*0.014*	*0.098*	*0.009*	*0.124*
	y_2	*0.028*	*0.164*	*0.025*	*0.265*
	y_3	*0.017*	*0.114*	*0.015*	*0.189*
	y_4	*0.018*	*0.212*	*0.025*	*0.339*
GSGP-LS	y_1	*0.001*	*0.014*	*0.001*	*0.019*
	y_2	*0.005*	*0.023*	*0.006*	*0.031*
	y_3	*0.003*	*0.019*	*0.003*	*0.038*
	y_4	*0.005*	*0.045*	*0.009*	*0.049*

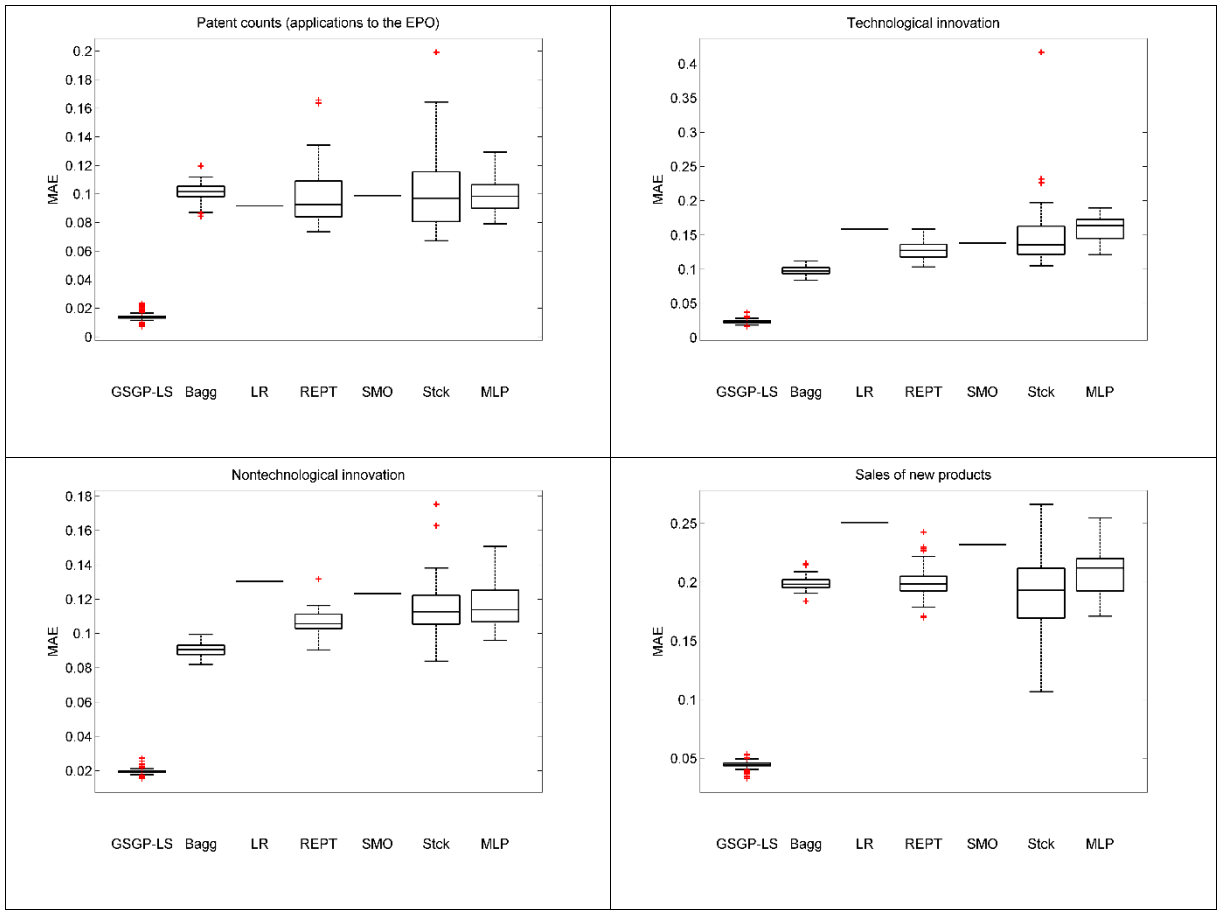
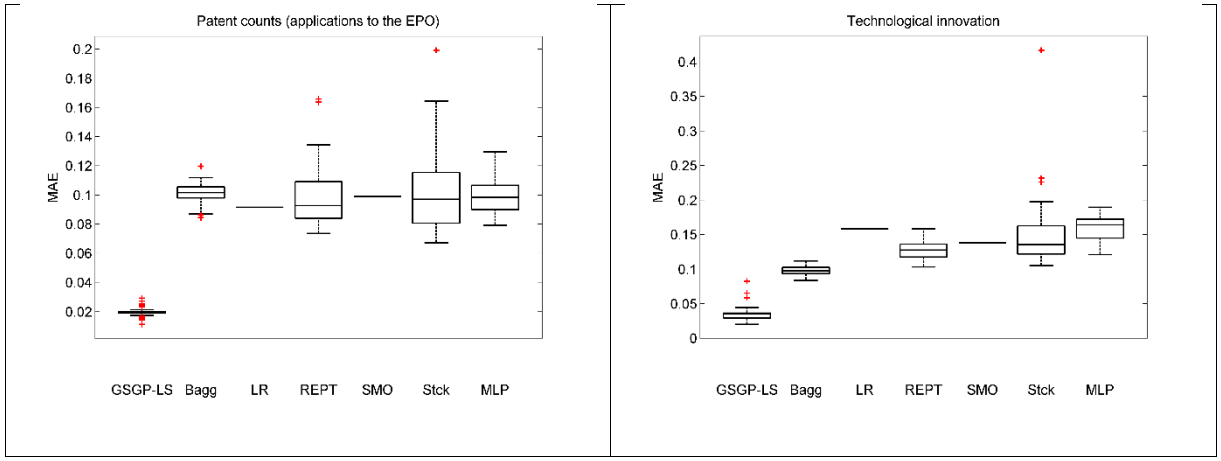


Figure 4 - Boxplots of mean absolute error (test fitness) for the 50 runs of the considered machine learning techniques on *Experiment 1*. On each box, the central mark is the median, the edges of the box are the 25th and 75th percentiles, and the whiskers extend to the most extreme data points not considered outliers.



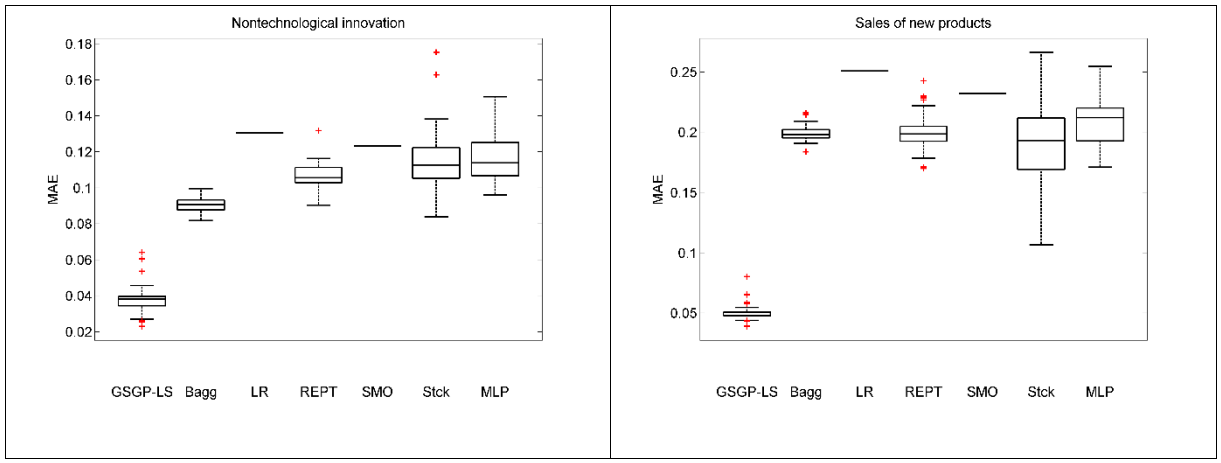


Figure 5 - Boxplots of mean absolute error (test fitness) for the 50 runs of the considered machine learning techniques on *Experiment 2*. On each box, the central mark is the median, the edges of the box are the 25th and 75th percentiles, and the whiskers extend to the most extreme data points not considered outliers.

To assess the statistical significance of the results obtained and to further corroborate the previous qualitative analysis, a set of tests was performed.

As a first step, the Lilliefors test was applied to test if the data comes from a normal distribution. The result of the test, performed with a significance level of 5%, suggested that the alternative hypothesis (data does not come from a normal distribution) cannot be rejected.

Thus, a rank-based statistic was used. More in detail, the Mann-Whitney U-test was performed with the null hypothesis that the samples have equal medians.

As for the previous test, a significance level of 5% with a Bonferroni correction was used. The *p*-values returned by the Mann-Whitney test (not reported in the manuscript) show that the differences in terms of median error between GSGP-LS and all the other machine learning techniques considered are statistically significant.

7. Conclusion and Future Directions

This study argued that the proposed GP-based model is more appropriate to model intrinsic complex and non-linear character of regional innovation performance than traditional statistical and machine learning forecasting models. The results of this study indicate that the GP-based model significantly outperforms other forecasting models in terms of test error. These results also suggest that the proposed forecasting model not only provides a good solution on training data but it also avoids overfitting due to the effective exploration-exploitation trade-off. In addition, this dominant performance was present for a variety of

innovation output forecasts, indicating its possible use in forecasting related innovation performance measures. This suggests that the proposed model addressed the problems related to the complexity and dynamic changes in regional innovation systems. Thus, it represents an accurate and reliable forecasting tool to support decision making in R&D and innovation policy. Specifically, the forecasting model could be further used to the simulation and development of targeted interventions aimed at improving the performance of regional innovation systems.

Our findings in this study were limited in several ways. The most important limitation was the availability of regional-level data as the results from the Community Innovation Survey are released with several years of delay. Further research might also explore the forecasting capacity of the model in different socio-economic and socio-political conditions and particular economic sectors, such as manufacturing or services. This is particularly important with regard to the current implementation of smart specialization policies. We also believe that the proposed forecasting model can be generalised and provide guidelines to the development of GP-based forecasts in a wide range of related socio-economic applications.

8. Acknowledgements

We gratefully acknowledge the help provided by constructive comments of the anonymous referees. This work was supported by the scientific research project of the Czech Sciences Foundation Grant No: 17-11795S.

9. References

[1] Carayannis EG, Grigoroudis E, Goletsis Y. A multilevel and multistage efficiency evaluation of innovation systems: A multiobjective DEA approach. Expert Syst Appl 2016;62:63–80. doi:10.1016/j.eswa.2016.06.017.

[2] Hajek P, Henriques R, Hajkova V. Visualising components of regional innovation systems using self-organizing maps—Evidence from European regions. Technol Forecast Soc Change 2014;84:197–214. doi:10.1016/j.techfore.2013.07.013.

[3] Lau AKW, Lo W. Regional innovation system, absorptive capacity and innovation performance: An empirical study. Technol Forecast Soc Change 2015;92:99–114. doi:10.1016/j.techfore.2014.11.005.

[4] Freeman C. Continental, national and sub-national innovation systems—complementarity and economic growth. Res Policy 2002;31:191–211.

doi:10.1016/S0048-7333(01)00136-6.

[5] Tödtling F, Trippl M. One size fits all? Res Policy 2005;34:1203–19. doi:10.1016/j.respol.2005.01.018.

[6] Hollanders H, Es-Sadki N, Kanerva M. European Innovation Scoreboard - European Commission. Brussels: European Commission; 2016. doi:10.2873/84537.

[7] Evangelista R, Iammarino S, Mastrostefano V, Silvani A. Measuring the regional dimension of innovation. Lessons from the Italian Innovation Survey. Technovation 2001;21:733–45. doi:10.1016/S0166-4972(00)00084-5.

[8] Audretsch DB, Feldman MP. Knowledge spillovers and the geography of innovation. Handb Reg Urban Econ 2004;4:2713–39. doi:10.1016/S1574-0080(04)80018-X.

[9] Naz A, Niebuhr A, Peters JC. What's behind the disparities in firm innovation rates across regions? Evidence on composition and context effects. Ann Reg Sci 2015;55:131–56. doi:10.1007/s00168-015-0694-9.

[10] Wang TY, Chien SC. Forecasting innovation performance via neural networks - A case of Taiwanese manufacturing industry. Technovation 2006;26:635–43. doi:10.1016/j.technovation.2004.11.001.

[11] Chien SC, Wang TY, Lin SL. Application of neuro-fuzzy networks to forecast innovation performance - The example of Taiwanese manufacturing industry. Expert Syst Appl 2010;37:1086–95. doi:10.1016/j.eswa.2009.06.107.

[12] Hajek P, Henriques R. Modelling innovation performance of European regions using multi-output neural networks. PLoS One 2017;12:e0185755. doi:10.1371/journal.pone.0185755.

[13] de la Paz-Marín M, Campoy-Muñoz P, Hervás-Martínez C. Non-linear multiclassifier model based on Artificial Intelligence to predict research and development performance in European countries. Technol Forecast Soc Change 2012;79:1731–45. doi:10.1016/j.techfore.2012.06.001.

[14] Hung SC, Tu MF. Is small actually big? the chaos of technological change. Res Policy 2014;43:1227–38. doi:10.1016/j.respol.2014.03.003.

[15] Samara E, Georgiadis P, Bakouros I. The impact of innovation policies on the performance of national innovation systems: A system dynamics analysis. Technovation 2012;32:624–38. doi:10.1016/j.technovation.2012.06.002.

[16] Castelli M, Manzoni L, Popovič A. An artificial intelligence system to predict quality of service in banking organizations. Comput Intell Neurosci 2016;2016:1–7. doi:10.1155/2016/9139380.

[17] Vanneschi L. An introduction to geometric semantic genetic programming. Stud Comput Intell 2017;663:3–42. doi:10.1007/978-3-319-44003-3_1.

[18] Castelli M, Trujillo L, Vanneschi L, Silva S, Z-Flores E, Legrand P. Geometric Semantic Genetic Programming with Local Search. Proc. 2015 Genet. Evol. Comput. Conf. - GECCO '15, 2015, p. 999–1006. doi:10.1145/2739480.2754795.

[19] Cooke P. Knowledge economies. Clusters, learning and cooperative advantage. London: Routledge; 2002. doi:10.4324/9780203445402.

[20] Rodriguez-Pose A, Crescenzi R. Research and development, spillovers, innovation systems, and the genesis of regional growth in Europe. Reg Stud 2008;42:51–67. doi:10.1080/00343400701654186.

[21] Huggins R, Thompson P. Entrepreneurship, innovation and regional growth: a network theory. Small Bus Econ 2015;45:103–28. doi:10.1007/s11187-015-9643-3.

[22] Annoni P, Dijkstra L, Gargano N. EU regional competitiveness index. Luxemburg: European Commission; 2017. doi:10.2776/94425.

[23] Frenz M, Ietto-Gillies G. The impact on innovation performance of different sources of knowledge: Evidence from the UK Community Innovation Survey. Res Policy 2009;38:1125–35. doi:10.1016/j.respol.2009.05.002.

[24] Asheim BT, Boschma R, Cooke P. Constructing regional advantage: Platform policies based on related variety and differentiated knowledge bases. Reg Stud 2011;45:893–904. doi:10.1080/00343404.2010.543126.

[25] Hájková V, Hájek P. Efficiency of knowledge bases in urban population and economic growth - Evidence from European cities. Cities 2014;40:11–22. doi:10.1016/j.cities.2014.04.001.

[26] Navarro M, Gibaja JJ, Bilbao-Osorio B, Aguado R. Patterns of innovation in EU-25 regions: a typology and policy recommendations. Environ Plan C Gov Policy 2009;27:815–40.

[27] Schürmann C, Talaat A. The European Peripherality Index. 42nd Congr. Eur. Reg. Sci. Assoc., 2002, p. 27–31.

[28] Vila LE, Cabrer B, Pavía JM. On the relationship between knowledge creation and economic performance. Technol Econ Dev Econ 2015;21:539–56.

[29] Ketata I, Sofka W, Grimpe C. The role of internal capabilities and firms' environment for sustainable innovation: evidence for Germany. R&D Manag 2014:n/a-n/a. doi:10.1111/radm.12052.

[30] Acs ZJ, Anselin L, Varga A. Patents and innovation counts as measures of regional

production of new knowledge. Res Policy 2002;31:1069–85. doi:10.1016/S0048-7333(01)00184-6.

[31] OECD. Frascati Manual: Proposed Standard Practice for Surveys on Research and Experimental Development. vol. 6th. 2002. doi:10.1787/9789264199040-en.

[32] Brenner T, Broekel T. Methodological issues in measuring innovation performance of spatial units. Ind Innov 2011;18:7–37. doi:10.1080/13662716.2010.528927.

[33] Bottazzi L, Peri G. Innovation and spillovers in regions: Evidence from European patent data. Eur Econ Rev 2003;47:687–710. doi:10.1016/S0014-2921(02)00307-0.

[34] Koza JR. Genetic programming: on the programming of computers by means of natural selection (Vol. 1). vol. 33. 1992. doi:10.1016/0303-2647(94)90062-0.

[35] A.E. Eiben JES, Eiben AE, Smith J. Introduction to Evolutionary Computing. Springer 2003. doi:10.1162/evco.2004.12.2.269.

[36] Moraglio A, Krawiec K, Johnson CG. Geometric semantic genetic programming. Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 7491 LNCS, 2012, p. 21–31. doi:10.1007/978-3-642-32937-1_3.

[37] Poli R, Langdon WB, McPhee NF. A Field Guide to Genetic Programing. 2008.

[38] Vanneschi L, Castelli M, Silva S. A survey of semantic methods in genetic programming. Genet Program Evolvable Mach 2014;15:195–214. doi:10.1007/s10710-013-9210-0.

[39] Castelli M, Silva S, Vanneschi L. A C++ framework for geometric semantic genetic programming. Genet Program Evolvable Mach 2015;16:73–81. doi:10.1007/s10710-014-9218-0.

[40] Chen X, Ong YS, Lim MH, Tan KC. A multi-facet survey on memetic computation. IEEE Trans Evol Comput 2011;15:591–607. doi:10.1109/TEVC.2011.2132725.

[41] Chen H, Zhang G, Zhu D, Lu J. A patent time series processing component for technology intelligence by trend identification functionality. Neural Comput Appl 2015;26:345–53. doi:10.1007/s00521-014-1616-y.

[42] Topchy A, Punch WF. Faster Genetic Programming based on Local Gradient Search of Numeric Leaf Values. Proc Genet Evol Comput Conf GECCO2001 2001:155–62.

[43] Zhang M, Smart W. Genetic Programming with Gradient Descent Search for Multiclass Object Classification. Genet. Program. 7th Eur. Conf. EuroGP 2004, Proc., vol. 3003, 2004, p. 399–408. doi:doi:10.1007/978-3-540-24650-3_38.

[44] Graff M, Pena R, Medina A, Ieee. Wind Speed Forecasting using Genetic Programming. 2013.

[45] Smart W, Zhang M. Continuously Evolving Programs in Genetic Programming using Gradient Descent. Proc. Second Asian-Pacific Work. Genet. Program., 2004, p. 16pp.

[46] Kommenda M, Kronberger G, Winkler S, Affenzeller M, Wagner S. Effects of constant optimization by nonlinear least squares minimization in symbolic regression. Proceeding Fifteenth Annu Conf Companion Genet Evol Comput Conf Companion - GECCO '13 Companion 2013:1121. doi:10.1145/2464576.2482691.

[47] Z-Flores E, Trujillo L, Schuetze O, Legrand P. Evaluating the Effects of Local Search in Genetic Programming. EVOLVE - A Bridg. between Probab. Set Oriented Numer. Evol. Comput. V, vol. 288, 2014, p. 213–28. doi:doi:10.1007/978-3-319-07494-8_15.

[48] Sorensen DC. Newton's Method With a Model Trust Region Modification. SIAM J Numer Anal 1982;19:409–26. doi:10.1137/0719026.

[49] Vanneschi L, Silva S, Castelli M, Manzoni L. Geometric Semantic Genetic Programming for Real Life Applications. Genet. Program. Theory Pract. XI, 2013, p. 191–209. doi:doi:10.1007/978-1-4939-0375-7_11.

[50] Keijzer M. Improving Symbolic Regression with Interval Arithmetic and Linear Scaling. Genet Program Proc EuroGP2003 2003;2610:70–82. doi:10.1007/3-540-36599-0_7.

[51] Moraglio A, Mambrini A. Runtime Analysis of Mutation-Based Geometric Semantic Genetic Programming for Basis Functions Regression. Gecco'13 Proc 2013 Genet Evol Comput Conf 2013:989–96. doi:doi:10.1145/2463372.2463492.

[52] Krawiec K, O'Reilly U-M. Behavioral Programming: A Broader and More Detailed Take on Semantic GP. Gecco 2014:935–42. doi:10.1145/2576768.2598288.

[53] Qi M, Zhang GP. Trend time-series modeling and forecasting with neural networks. IEEE Trans Neural Networks 2008;19:808–16. doi:10.1109/TNN.2007.912308.

[54] Tripepi G, Jager KJ, Dekker FW, Zoccali C. Linear and logistic regression analysis. Kidney Int 2008;73:806–10. doi:10.1038/sj.ki.5002787.

[55] Rumelhart DE, Hinton GE, Williams RJ. Learning representations by back-propagating errors. Nature 1986;323:533–6. doi:10.1038/323533a0.

[56] Elomaa T, Kääriäinen M. An analysis of reduced error pruning. J Artif Intell Res 2001;15:163–87. doi:10.1613/jair.816.

[57] Basak D, Pal S, Patranabis DC. Support vector regression. Neuronal Inf Process - Lett Rev 2007;11:203–24. doi:10.4258/hir.2010.16.4.224.

[58] Breiman L. Stacked regressions. Mach Learn 1996;24:49–64. doi:10.1007/BF00117832.

[59] Breiman L. Bagging predictors. Mach Learn 1996;24:123–40. doi:10.1007/BF00058655.

[60] Frank E, Hall MA, Witten IH. The WEKA Workbench. Morgan Kaufmann, Fourth Ed., 2016, p. 553–71.

Appendix 1: Descriptive statistics (Mean±St.Dev.) on input and output attributes

	2004	2006	2008	2010	2012
x_1	0.268±0.173	0.297±0.184	0.300±0.179	0.273±0.178	-
x_2	0.568±0.202	0.568±0.202	0.608±0.196	0.665±0.191	-
x_3	0.557±0.222	0.557±0.222	0.607±0.237	0.610±0.241	-
x_4	0.762±0.177	0.762±0.177	0.623±0.204	0.346±0.269	-
x_5	0.509±0.176	0.509±0.176	0.426±0.193	0.362±0.213	-
x_6	0.486±0.222	0.486±0.222	0.554±0.230	0.621±0.241	-
x_7	0.440±0.219	0.451±0.219	0.463±0.205	0.377±0.205	-
x_8	0.397±0.178	0.406±0.166	0.407±0.163	0.364±0.165	-
x_9	0.217±0.162	0.153±0.118	0.226±0.160	0.221±0.155	-
x_{10}	0.499±0.105	0.462±0.108	0.516±0.121	0.359±0.120	-
x_{11}	0.404±0.197	0.417±0.199	0.436±0.193	0.406±0.203	-
x_{12}	0.475±0.204	0.496±0.203	0.492±0.206	0.508±0.177	-
x_{13}	0.499±0.157	0.522±0.158	0.556±0.167	0.493±0.177	-
x_{14}	0.275±0.205	0.274±0.194	0.271±0.196	0.269±0.195	-
x_{15}	0.534±0.243	0.534±0.243	0.534±0.243	0.534±0.243	-
x_{16}	0.055±0.082	0.055±0.081	0.052±0.079	0.055±0.081	-
x_{17}	0.373±0.191	0.373±0.191	0.373±0.191	0.373±0.191	-
x_{18}	0.343±0.159	0.360±0.159	0.384±0.192	0.417±0.239	-
y_1	0.449±0.202	0.448±0.192	0.449±0.184	0.291±0.178	0.345±0.196
y_2	0.447±0.182	0.444±0.197	0.474±0.219	0.475±0.268	0.449±0.181
y_3	0.446±0.145	0.493±0.197	0.439±0.218	0.411±0.209	0.350±0.158
y_4	0.449±0.161	0.421±0.178	0.471±0.159	0.461±0.140	0.320±0.159