# Forecasting Influenza-Like Illness Rates Using Online Search Data

Ahmed Elherazy, Samuel Bouilloud, Walter Wu, Wiryawan Mehanda, and Yuxuan Li
Dpt. of Computer Science
University College London, London, UK

***Abstract*—Influenza is a recurrent seasonal illness that affects millions and kills hundreds-of-thousands annually. Tracking rates of influenza-like illnesses (ILIs) across populations are therefore of interest to medical and academic institutions. As the Health Protection Agency (HPA) releases its official weekly ILI rate data with a 1-2 week reporting lag, efforts have been made to perform forecasting and real-time tracking (nowcasting) of ILI rates using alternative online data sources. Daily Google search query data has been shown to be a relatively accurate predictor of the current true ILI rate when applied in various supervised machine learning and statistical models. This study presents five distinct supervised models for ILI nowcasting; Lasso, Random Forest Regression, Gaussian Processes, Support Vector Regression, and Multilayer Perceptron, trained on daily Google query frequency and historical ILI data. Ultimately, a novel ensemble approach was constructed from the aforementioned constituent models, which resulted in the most accurate and robust prediction for ILI nowcasts. Two approaches were studied for 1-week and 2-week ahead ILI forecasting, being Multilayer Perceptron and Autoregressive Integrated Moving Average, both having competitive performance in different test metrics.**

## I. Introduction

Influenza-like illness (ILI) is a seasonal and non-seasonal disease that presents several symptoms, commonly fever, cough, or sore throat. Every year, such illness strikes the world. Globally, these annual epidemics are estimated to cause 3 to 5 million severe cases, and about 290,000 to 650,000 deaths [1]. Prediction of the rates of such disease is crucial to public health officials to allow them to prepare and equip properly, develop preventing strategies and countermeasures, which can help avoid terrific pandemics and save lives. Currently, traditional surveillance systems to continuously monitor ILI rates in the population are done by organizations such as the Center of Disease Control and Prevention (CDC) in the United States of America and the Health Protection Agency (HPA) in the United Kingdom. Such measurements are provided by collecting and compiling data from various health care providers who report on the percentage of patients seen who exhibit ILI symptoms. CDC's ILI reports are available to health care providers with a known delay of 7-14 days. This means that once the reports are released, the data is already 1-2 weeks old. This time lag prevents optimal decision-making and instant preparation, such as allocation resources like vaccines, staff, and other commodities across the country. Hence, the availability of real-time rates of the prevalence of the disease in the population is critical to control outbreaks and reduce its impact.

Thus, many researchers attempted to solve the time lag problem by providing accurate real-time estimates (nowcast) and future estimates (forecast) of ILI activity using a combination of data and methods. Statistical and mechanistic disease models (a model that simulates the spreading of a disease), alongside with other machine learning techniques were proposed and compared to find the best performing and robust technique. Further, combinations of these techniques were proposed too to benefit from their distinct strength and overcome their weaknesses. These methods used multiple different data such as meteorological, demographic, and epidemiological data.

In recent years, Internet-based data drew the attention of the community and showed significant potential in the prediction of current trends and real-life patterns such as unemployment, stock market prices, political opinions, and house prices. Internet-based data are constantly generated as millions of users access Internet-based services every second, thanks to smartphones. These services nowadays became very powerful and embed in their streams extraordinarily valuable information about human behaviour. Such data attracted the researchers' interest in the public health area as well. Using services such as Google [2], [3], [4], [5], [6], [7], Wikipedia [8], and Twitter [9], [10], [11], [12], [13], researchers have been able to accurately nowcast and forecast ILI activity. Google Flu Trend (GFT) [2], a pioneer influenza web-service, developed in 2008 by Google to predict ILI activity in 25 countries based on flu-related Google search queries. Such service was initially hailed as a success, but several doubts have been raised after it mispredicted the non-seasonal 2009 H1N1 pandemic and 2013 influenza season. It was discontinued in 2015, and its data was made public and free to use. Even though GFT did not end up being a success, alternative solutions have been proposed to overcome GFT's limitations. To accomplish the latter, various approaches have been proposed for nowcasting and forecasting ILI activity. They are typically based on linear models such as linear regression or non-linear models such as Gaussian Processes (GP) or even neural networks, for instance, multilayer perceptron (MLP). For forecasting, commonly used models were based on variations of autoregressive models such as autoregressive moving average (ARMA) or autoregressive integrated moving average (ARIMA).

In this study, we aim to develop and evaluate nowcasting and forecasting methods for ILI rates in the UK. We used ILI rate time series from the Public Health England and Google

search queries frequencies to train and test our models. Since our query dataset was unfiltered, including unrelated queries to our study, we had to do a query selection step to address this issue using methods such as manual selection, correlation, and regularization. To develop ILI nowcasts, we considered six different modelling approaches: Least Absolute Shrinkage and Selection Operator (LASSO), Gaussian Process (GP), Support Vector Regression (SVR), Random Forest Regression (RFR), Multilayer Perceptron (MLP) and finally, an ensemble model using these last five approaches. For forecasting, we provide forecasts for one-week and two-weeks ahead using two approaches: ARIMA and MLP.

As a next step, the remainder of this paper is organized in four major sections: Methods and Material, Analysis, Discussion and Limitations, and Conclusion and Future Work. In the Methods and Material section, we will present the data used, the pre-processing methods applied such data, the metrics used to evaluate our models, and the models we developed in this study. Afterwards, in the Analysis section, we will state and compare the results we obtained from our models, to then discuss and explain such results and demonstrate the limitations of our study. Finally, we will conclude our paper and consider future possibilities.

## II. Materials and Methods

### A. Dataset Description

In order to adequately predict ILI rates in the UK, we have access to several data about the disease. First of all, we have the historical ILI rates in the United Kingdom, from 24/08/2005 to 23/08/2017, giving us 12 complete influenza seasons to study (available in Figure 1). These rates represent the number of infected out of 100,000 people and are obtained each week from the feedback of health practitioners around the country. As stated before, our main problem here is that these rates arrive with a week delay, making the nowcast task crucial in order to accurately forecast ILI rates. It is also important to note that, as the original data is weekly, the rates were linearly interpolated to give us daily data, and this might have its impact on the results.

On the other hand, we also have a set of 1,000 google queries rates for the same date interval. These queries were chosen using Google Trends, by recursively searching for the most correlated queries, and their rates represent the number of searches out of 10 million. As we could initially imagine, a large amount of these queries don't relate to flu, making the query selection process a crucial step in our research. Moreover, just like the ILI rates, this data is also very biased. Indeed, it is provided by Google without any more information about its precision but, more importantly, rates below an unknown threshold are rounded to 0. This gives us rates with a lot of spikes, justifying our choice for a heavy pre-processing of the data. Finally, in contrast with the ILI rates, this type of data is available each day with no delay, making it a very adequate feature set for the nowcasting task.

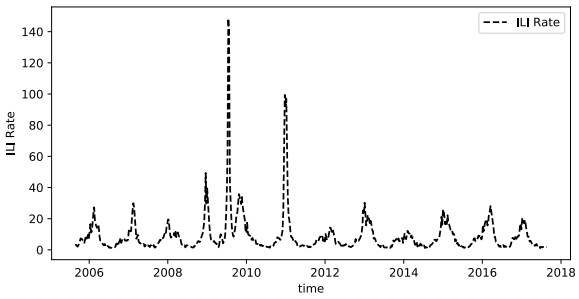


Fig. 1: ILI Rates from 2005 to 2017

### B. Data Selection

Our query selection follows a very carefully thought and tested process. First of all, to remove well-represented queries (i.e. usually also strongly used in the winter, thus very correlated to flu), we filter the 1,000 original ones with 56 chosen keywords, like 'pneumonia', 'hayfever' or even 'bronchitis' for example. This is the only step of this process that could not be easily automated, as we picked the keywords by hand, by looking for unrelated queries or very uncorrelated queries like "hayfever", as we can see in Figure 2. That figure shows the contrast between a very correlated query, that appears only during influenza season with quite the same intensity, and one the most uncorrelated queries of our set, appearing mainly out of the flu season. After doing that first filtering, for each of the test years, we rank the queries by their Pearson correlation to the true ILI rate, in the years prior to that (e.g. in 2005-2014 for the 2014/15 test year), and remove all the negatively correlated queries. These first two steps allow us to already remove around half the queries for each test year, but unrelated or inefficient keywords are still very present. The last step of our query selection process involves an L1 regularized linear model (LASSO), that adds a penalty to features in the input vector, resulting in some of them having 0 weight. Using this type of model, we train it on the last 2 years with the current set of around 500 queries as input and the true ILI rates as output. Thus, we can get the set of input features with non-zero weight, giving us our final set of queries per test year. These sets are of length 186, 308 and 372, for 2014/15, 2015/16 and 2016/17 respectively.

### C. Data Preprocessing

As stated in the data description, the rates of google queries we have are very inaccurate, present spikes and 0 values where they should not be. Using this data results in very inconstant predictions, that adds substantial error, and to avoid this, we decided to smooth the queries rates. Indeed, after testing several window sizes and weightings with an L2 linear regularized model (Ridge), we ended with a 5-day smoothing of the rates, using an average of the past 4 days with the current day. This measure removed the spikes and smoothed the curve, leading to improved error metrics and clearer visualizations to analyse.

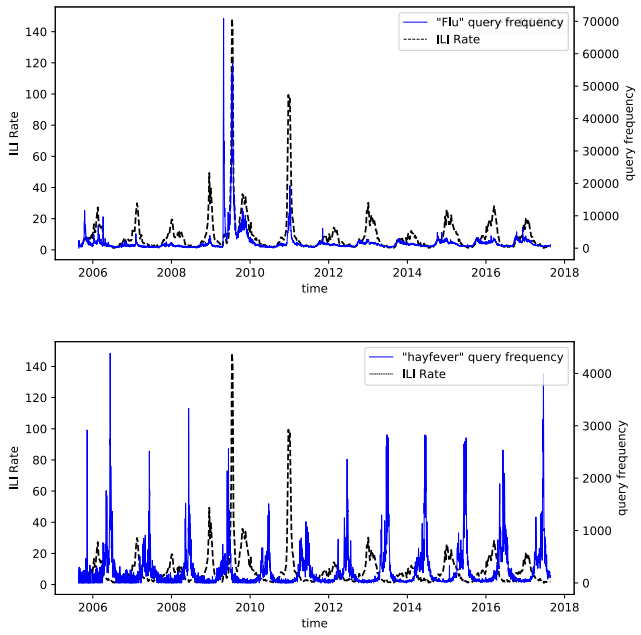


Fig. 2: ILI rates and query frequencies of "flu" and "hayfever", from 2005 to 2017

### D. Incorporating Historical ILI

All nowcasting models in this study have identical input variables. In addition to the Google query frequencies selected by the aforementioned query selection method, the last 30 days of available historical ILI rates prior to the date of prediction are also given as exogenous variables. For any given prediction date, we assume a 7-day reporting delay in the data made available by the HPA. This hence incorporates an auto-regressive component to our predictive models, an approach which previous literature has shown to provide improvements on ILI nowcasting result [3], [7].

The input data to the forecasting models consists of two parts: the first part is the 30 days historical ILI rates available at the time of the forecast, the second part is 7 days of nowcasted ILI rates covering the report delay. The nowcasted ILI rates are obtained using the Ensemble nowcast model, a combination of our best models and considered to be the most accurate and reliable nowcast model based on our analysis.

### E. Models

*1)* ***LASSO (Linear Regression)****:* Linear regression models have great simplicity and explainability, and they tend to achieve great performance on simple linear tasks. Since the number of queries is greatly reduced by filtering out unrelated ones, and the dataset is interpolated to produce more data points, the number of data points is sufficient to support an expressive linear model that keeps individual query frequencies separated. This model can be written as $y = \beta + \mathbf{w_x^T x} + \mathbf{w_y^T y_{old}} + \epsilon$, where $\mathbf{x}$ is a vector of query frequencies for target day, $\mathbf{y_{old}}$ is a vector historical ILI rates from 7 days before current date to 30 days before, $\beta$ and $\mathbf{w_x}$ $\mathbf{w_y}$ are intercepts and weights to learn and $\epsilon$ is independent noise.

Although this model does not suffer from under-determinant due to a sufficient number of data points, applying regularization can still effectively reduce the influence of noise in the data and prevent overfitting. This is particularly important as a large number of our data points are interpolated. An L1 regularization term is thus appended to this model, and the objective function becomes:

$$\arg\min_{w,\beta}\left(\sum_{t=1}^{T}(\mathbf{w_x^T x}_t + \mathbf{w_y^T x}_t + \beta - y_t) + \alpha(||\mathbf{w_x}|| + ||\mathbf{w_y}||)\right)$$

The L1 term $\alpha(||\mathbf{w_x}|| + ||\mathbf{w_y}||)$ increases and penalizes the loss function when the model has large weights, which is usually an indication of overfitting. In order to optimize the model performance, the value of $\alpha$ has been tuned over a large range to identify the maxima. [14]

*2)* ***Gaussian Processes (GPs)****:* The Gaussian Process Regression is a Bayesian approach to machine learning, that sets a probability distribution on all possible parameters. By computing prior and posterior distributions, the model is capable of adjusting to the dataset and, in our case, output an ILI rate from a set of Google queries' frequencies. Indeed, when observing the dataset (i.e. training phase), it relocates the probabilities by using Bayes' Rule, to fit to the given input/output. [14]

In a Gaussian Process Regressor, the first distribution assumed is the Gaussian process prior, given as a mean function and a covariance function, also known as the kernel. In addition, to adapt to real-life datasets, Gaussian noise can be added to this prior distribution. For our implementation, we tested several kernels and decided to always use the dot product function. Thus, the only hyperparameter that we chose to validate was the alpha, basically corresponding to the noise level.

Gaussian Processes are known to work very well with relatively small datasets, being very adequate for our work, as we have few data points to train the models on. However, their computation cost is very high and a full study on the multiple kernel functions would take too long. It's mainly for this reason that we decided to fix the kernel to a fairly simple one and advance on our work.

*3)* ***Random Forest Regression (RFR)****:* Random Forest is a supervised learning method applicable to both regression and classification problems, built upon an aggregation of decision trees [15]. When predicting with a set of values of the input features, a regression decision tree uses a cascade of conditional statements in assessing individual values of its input, traversing a binary-tree like structure with regression outputs in its leaf nodes. As there are only a finite amount of these leaf nodes, regression trees hence discretize the continuous output space typically found in regression problems.

RFR is a variation of regression tree bootstrap aggregating (bagging), where training data is separated into a number of bootstrapped sets to produce many variations of the regression

tree, whose regression is then collectively averaged to produce a single low-variance model. In addition, RFR randomly restricts a number of candidate features to be considered for branching in the construction of each regression tree. This has the benefit of randomly deterring "strong" predictors among input features, resulting in less-correlated regression trees for bagging.

There are numerous hyperparameters that can be tuned for RFR models, including the number of decision trees, splitting and height of decision trees, and the number of candidate features to be considered at branches. Realistically, there are too many hyperparameters to optimize for in validation of RFR, and hence we have decided on the latter to be tuned. We believe this to be the most influential parameter, as a lower number of candidate queries may reduce the prominence of input data that may cause a model to overfit outlier years (such as reducing the chances of 'swine flu' related queries to be chosen when the model is trained on 2009 ILI data). Given the random nature of the RFR model, this tuning is performed 10 times and averaged on two-prior influenza seasons to ensure a lower variance of our validation result.

*4) **Support Vector Regression (SVR)**:* Support Vector Regression (SVR) follows similar principles to the Support Vector Machine (SVM) Classification algorithm with minor variations. It is a learning algorithm that emerges from the optimization of a generalization bound on the $\epsilon$-insensitive loss. SVR allows us to flexibly set a certain threshold for the error tolerated while fitting our model to the data and finding an appropriate hyperplane. Such tolerance is defined by setting the $\epsilon$ parameter, called the maximum error, which represents a margin of tolerance. This is done by restricting our optimization problem with constraints as follows:

$$\min_{w,b,\zeta,\zeta'} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}(\zeta^{(i)} + \zeta'^{(i)})$$

subject to:

$$y^{(i)} - w \cdot x^{(i)} - b \leq \epsilon + \zeta^{(i)}$$

$$w \cdot x^{(i)} + b - y^{(i)} \leq \epsilon + \zeta'^{(i)}$$

$$\zeta^{(i)}, \zeta'(i) \geq 0$$

C and $\epsilon$ are hyperparameters that we can tune. C is a regularization parameter that controls the trade-off between the width of the margin and the tolerance for error. For instance, a larger C gives more weight to minimizing the error. Thus as C increases as we become less tolerant to points outside of the margin, and the complexity of our model increases. Thus, we chose only to tune the C hyperparameter as we believe it is the most impactful parameter and due to computational complexity.

Moreover, similarly to SVM, the SVR algorithm uses kernel methods to transform the data. Kernel methods seek to implicitly map the data into higher dimensional space to achieve higher accuracy. The most commonly used kernel functions are linear, polynomial, and radial basis function (RBF). After a comparison between the kernel functions, the polynomial kernel function was used in our model as it was the best performant. [14]

*5) **Multilayer Perceptron (MLP)**:* MLP is the most basic architecture of the Neural Network. It works by having several dense layers of neurons, each directly connected to neurons in the next layer by a certain weight and intercept. Each neuron also has an activation function that decides whether the neuron would fire or not and what is the output based on the given input. After applying weight and intercept of the connections, the output of a neuron is propagated to all neurons in the next layer and eventually, the last layer produces a single value as the result. In an MLP model, the first layer that's directly connected to input data is the input layer, the last layer is called the output layer, and layers between them are hidden layers.

Many aspects can affect the performance of MLP, including batch size, number of epochs, number of hidden layers, number of neurons, activation function and learning rate. Batch size is the amount of data input into the model for each update of the parameters. A smaller batch size allows faster training but the updates of parameters would fluctuate heavily. The number of epochs decides the extent of training. One epoch means that all available training data is used once to update the model. Having too few epochs prevents the model from learning while having too many epochs causes overfitting. In addition, increasing the number of hidden layers and neurons allows MLP to model more complex relationships, but more training data is needed. On the other hand, with no hidden layer, a MLP behaves exactly like a linear model. Finally, the activation function affects the activation of each individual neuron. The more typically used ones are *relu*, a linear activation function, *sigmoid* and *tanh*, which are non-linear activation functions.

Due to the complexity of tuning all hyperparameters, the batch size and number of epochs are not manually tuned. The batch size is fixed to full batch in training, meaning that all available training data is given to model for each update. This is practical because we only have approximately 3,500 data points for training. In order to determine the appropriate number of epochs, 10% of the training data are reserved as validation data and the technique of early stopping is applied. When the model's validation score does not improve within the previous 20 epochs, the training would be stopped and the parameters that give the best score are restored. [14]

*6) **Nowcast Model Ensemble**:* In addition to the various models described above, a final approach to nowcasting, in the form of a model ensemble, is investigated. Each of the aforementioned models presents distinct strengths and limitations which may result in varying degrees of nowcasting success throughout a single influenza season. For instance, while RFR excels in building models which exhibit complex relationships among its features, it does not take advantage of possible linearity in the data which may be captured more readily by regularised linear regression. Hence to accommodate this variance of model capabilities, we introduce an ensemble

approach which "averages" the respective ILI predictions of the four described models to produce a theoretically more stable result.

The resultant ILI predictions are ensembles using a weighted average approach based on the RMSE scores of each model on the two preceding influenza seasons, in accordance to the following formula:

$$y_{ensemble} = \left( \sum_{i=1}^{|m|} y_i * \frac{1}{RMSE_i} \right) * \frac{1}{\left( \sum_{i=1}^{|m|} y_i * \frac{1}{RMSE_i} \right)}$$

Where $y_i$ and $RMSE_i$ are the predicted ILI rates and $RMSE$ score of the $i^{th}$ model in the ensemble respectively. The above weighting method is chosen as a lower $RMSE$ indicates a better performing model, hence the coefficient $\frac{1}{RMSE}$ would assign higher weights to these models. The summation of weighted ILI predictions is then normalized on the sum of these coefficients.

While a number of different ensemble methods were considered for this study, such as a non-weighted average approach and squaring the weights to give even more prominence to well-performing methods, preliminary testing on the training set has favoured the weighted averaging formula described above, and hence it was selected. However, this paper will not delve into the details of this ensemble method selection.

*7)* ***Auto-Regression (ARIMA)****:* This model has many differences with the other, more classical machine learning models, as it is specially designed for time-series data. Indeed, although we included historical ILI data previously, autoregression uses regression analysis to predict a value given its past data. This type of model exists in various iterations, and we are here using an Autoregressive Integrated Moving Average version (ARIMA). It is composed by three parts, given as : Autoregressive part (AR), that can be a model on its own, computing a given variable regressing on its own prior values; Integration (I), that represents the differencing needed to make the time series stationary (i.e. values are replaced by the difference between current and past values); Moving average part (MA), that uses the dependency between a value and the residual error from a moving average model applied on lagged observations.

This type of autoregressive model is used with a rolling forecast, where we fit the model for each day of the year and make the corresponding forecasts (e.g. 7 or 14 days ahead). It mainly needs three hyperparameters, giving us the order of the model : p: the number of prior values to include in the model; d: the degree of differencing for the values we include; q: the size of the moving average window. [16]

The fact that true ILI rates arrive with a week delay affects the use of this type of model. Indeed, to avoid having to add a week of "uncertainty" when making our forecasts, we decided to cover the missing values with the outputs from our best nowcast model, the ensemble. Like that we can still make 7 days ahead forecast to predict the value in a week time, even if the use of nowcast involves some inevitable error. It is also important to note that autoregression is computationally very slow, as the rolling model needs to be re-fitted for each day of the year, and we will see that this affected our validation method.

### *F. Validation*

To validate model performance and tune hyperparameters, we decided to leave out the last two years of training data as validation data and use Mean Absolute Error (MAE) as the validation metric. We compared several validation methods such as cross-validation, two years and three years validation, by rigorously testing them with linear models like LASSO and Ridge. To do so, we validated each of the three tests with the three stated methods and compared them with chosen metrics. The two and three past years validation works by averaging the MAE over the previous years, for each hyperparameter set; and the k-fold cross validation by iteratively leaving one year out and averaging the score on all years. From these results, the chosen hyperparameters are the ones giving the best validation score. By doing this experiment, we concluded that two and three past years validation represented better the actual test year, as users' habits on querying online gradually change over time, and decided to stick with only two years for computational purposes.

The only model where the validation method is different is for the autoregressive model, ARIMA. Indeed, as it requires us to re-fit the model for each day of the year, doing it for the past two years for each hyperparameter would be computationally very demanding. To overcome that problem, we decided to use short-term previous data in a different way. First, to get the autoregressive term (p), we plot the partial autocorrelation over the past 2 years. As this gives us the relation between a value and its lags, we can decide until which day it is worth going back. Secondly, finding the moving average part (q) uses the autocorrelation plots, telling us how many prior values are required to remove autocorrelation in the stationarized series. Finally, the order of differencing (d) is the minimum order required to make the series stationary. Thus, we plot the autocorrelation for each order of differencing and take the first order where values reach 0 quite quickly. [16]

### *G. Metrics*

Multiple metrics including Mean Absolute Error (MAE), Root Mean Square Error (RMSE) and Pearson correlation are applied on the different years when testing in order to fully represent models' performance.

Our main metric is the MAE, as we use it in models' validation, revealing the general error between predictions and expected values. On the other hand, the RMSE penalizes more on large differences, which mainly tend to happen during the peaks of ILI rate and can be significant in real-world applications. Finally, the Pearson correlation measures the linear correlation between predicted and actual values, giving us an idea on how the predictions' curve follows the actual ILI curve.

## III. Analysis

### A. Nowcasting

| Test Year | Model | MAE | RMSE | Correlation |
|---|---|---|---|---|
| **2014** | **MLP** | 0.9383 | **1.4291** | **0.9791** |
| | **LASSO** | 1.2528 | 1.8068 | 0.9668 |
| | **Random Forest** | 1.0349 | 2.147 | 0.9606 |
| | **Gaussian Processes** | 1.2557 | 1.8112 | 0.9666 |
| | **SVR** | 1.1039 | 1.9043 | 0.9762 |
| | **Ensemble** | **0.9287** | 1.5806 | 0.9757 |
| **2015** | **MLP** | 1.3299 | 1.9221 | 0.9668 |
| | **LASSO** | 1.5920 | 2.0671 | 0.9661 |
| | **Random Forest** | 1.2460 | 1.8487 | 0.9681 |
| | **Gaussian Processes** | 1.3771 | 1.8018 | 0.9709 |
| | **SVR** | 1.2741 | 1.6818 | 0.9731 |
| | **Ensemble** | **1.1176** | **1.5097** | **0.9805** |
| **2016** | **MLP** | 0.9460 | 1.2739 | 0.9750 |
| | **LASSO** | 1.0547 | 1.3847 | 0.9692 |
| | **Random Forest** | 1.2520 | 1.9759 | 0.9675 |
| | **Gaussian Processes** | 1.2912 | 1.6727 | 0.9618 |
| | **SVR** | 1.4783 | 2.1017 | 0.9477 |
| | **Ensemble** | **0.8562** | **1.1977** | **0.9775** |
| **Average** | **MLP** | 1.0714 | 1.5417 | 0.9736 |
| | **LASSO** | 1.2998 | 1.7528 | 0.9674 |
| | **Random Forest** | 1.1776 | 1.9905 | 0.9654 |
| | **Gaussian Processes** | 1.3080 | 1.7619 | 0.9664 |
| | **SVR** | 1.2854 | 1.8959 | 0.9657 |
| | **Ensemble** | **0.9675** | **1.4293** | **0.9779** |

TABLE I: Nowcast Scores

Table 1 presents the performance of all five nowcast models tested on 2014-2015, 2015-2016, and 2016-2017 influenza seasons evaluated on MAE, RMSE, and Pearson correlation. An additional set of scores is produced as an average of the three test years as to provide a means to evaluate average model performance, as some models seem to perform better in certain test years than others. ILI predictions generated throughout three test years are then concatenated to generate the ILI time series shown in Figure 3.

On average, the weighted average ensemble model achieved the best performance across all metrics compared to its constituent models. In the second testing period (2015-2016) in particular, it outperforms all other approaches in both RMSE and MAE in an influenza season without a clear second-best model, Random Forest achieved better MAE results whereas Gaussian Processes achieved better RMSE. In the third testing period (2016-2017), however, the ensemble model only narrowly outperforms our MLP model with a margin of $< 3\%$ in MAE and $< 1\%$ in RMSE and correlation, making its performance less conclusive. This is particularly notable considering the results of the first testing period (2014-2015), where MLP outperforms the Ensemble model on RMSE and correlation by similarly small margins.

Examining the models individually reveals inconsistency in the performance of each approach. While MLP produces the second-best scores behind ensemble on average, it underperforms in the 2015-2016 test season on two metrics in comparison to RFR and GP. Additionally, a worst-performing model in our setting cannot be decisively determined as the RFR model resulted in the highest RMSE by a decent margin, while also well outperforming Lasso, GP and SVR in MAE with comparable correlation scores. GP on average achieved the worst MAE and correlation performance, although all its scores differ from Lasso by an extremely small margin ($< 1\%$). For this reason, none of these models is excluded in the computation of our ensemble, as they seem to provide varying strengths in prediction.

### B. Forecasting

| Test Year | Model | MAE | RMSE | Correlation |
|---|---|---|---|---|
| **2014** | **Shift** | 2.0013 | 2.8874 | 0.9078 |
| | **MLP** | **1.4932** | 2.5390 | 0.9290 |
| | **ARIMA** | 1.5595 | **2.4254** | **0.9452** |
| **2015** | **Shift** | 2.2885 | 3.2641 | 0.8971 |
| | **MLP** | 2.0953 | 2.9281 | 0.9233 |
| | **ARIMA** | **1.7195** | **2.3955** | **0.9450** |
| **2016** | **Shift** | 1.6112 | 2.3913 | 0.9063 |
| | **MLP** | **1.5433** | **2.2023** | 0.9200 |
| | **ARIMA** | 1.5698 | 2.2573 | **0.9257** |
| **Average** | **Shift** | 1.9670 | 2.8476 | 0.9037 |
| | **MLP** | 1.7106 | 2.5565 | 0.9241 |
| | **ARIMA** | **1.6163** | **2.3594** | **0.9386** |

TABLE II: 7 Days Ahead Forecast Scores

| Test Year | Model | MAE | RMSE | Correlation |
|---|---|---|---|---|
| **2014** | **Shift** | 2.7068 | 3.8008 | 0.8403 |
| | **MLP** | **2.1244** | 3.4757 | 0.8642 |
| | **ARIMA** | 2.2285 | **3.3980** | **0.8940** |
| **2015** | **Shift** | 3.0569 | 4.3952 | 0.8135 |
| | **MLP** | **2.4495** | 3.6616 | 0.8679 |
| | **ARIMA** | 2.5641 | **3.4165** | **0.8825** |
| **2016** | **Shift** | 2.1144 | 3.0230 | 0.8501 |
| | **MLP** | 2.0955 | 2.9643 | 0.8469 |
| | **ARIMA** | **2.0298** | **2.9462** | **0.8758** |
| **Average** | **Shift** | 2.6260 | 3.7397 | 0.8346 |
| | **MLP** | **2.2232** | 3.3672 | 0.8597 |
| | **ARIMA** | 2.2741 | **3.2536** | **0.8841** |

TABLE III: 14 Days Ahead Forecast Scores

Tables 2-3 and figure 4 present the results of forecasting models for 7 days ahead and 14 days ahead, evaluated on MAE, RMSE and correlation. We also present a baseline, the shift model, where each target ILI rate is the latest available historical ILI rate (e.g. the rate 7 or 14 days before).

MLP and ARIMA show a competing performance and both outperform the baseline model across all metrics and testing seasons. Examining individual metrics reveals that ARIMA consistently achieves the best correlation, while MLP occasionally suppresses ARIMA in terms of MAE and RMSE, especially with 14 days ahead forecast, where MLP exceeds ARIMA by 0.05 in terms of average MAE.

By comparing the scores among each testing season, it is notable that all models perform significantly worse in the second period (2015 2016). Figure 4 reveals that it is due to the severe misprediction of the peak between Jan 2016 and May 2016.

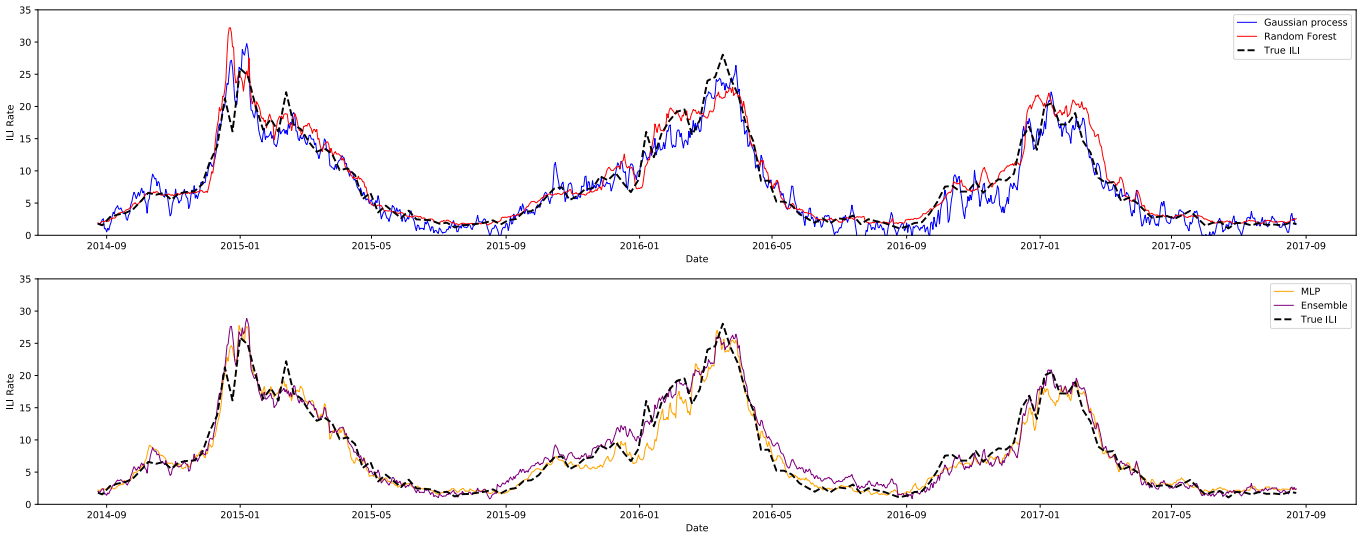


Fig. 3: Nowcasting model results. Nowcasting predictions for Gaussian Processes and Random Forest Regression (above), MLP and Ensemble (below)

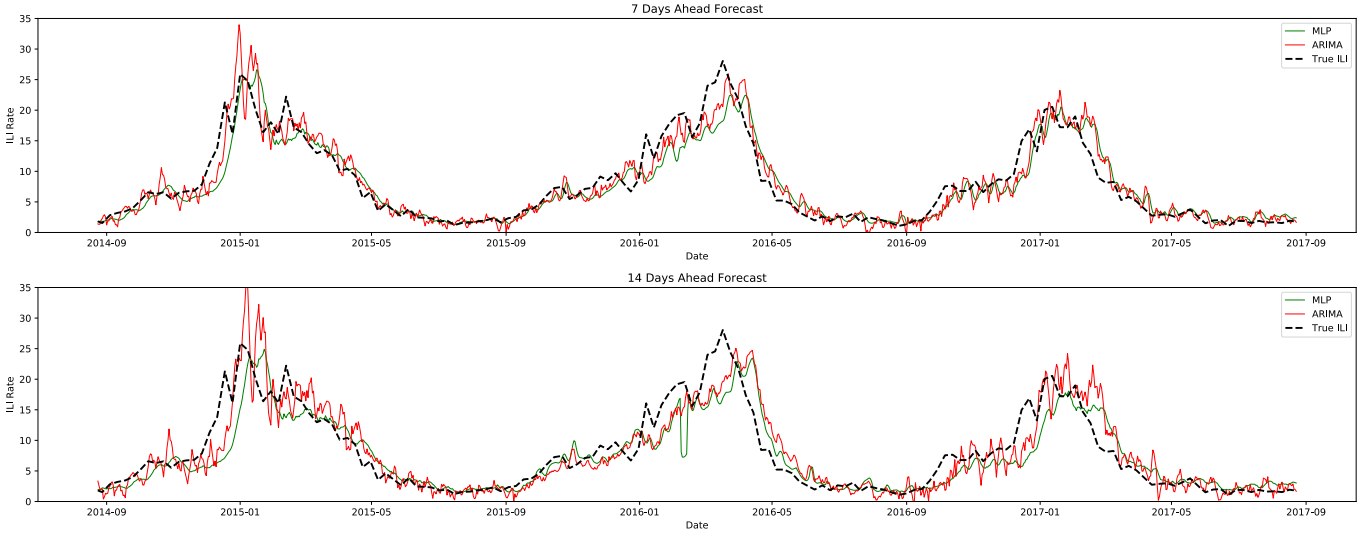


Fig. 4: Forecasting results of ARIMA and MLP models. The first plot shows 7 days ahead forecast and the second one 14 days ahead forecast.

## IV. Discussion & Limitations

### A. Methodology Discussion

One of the main steps in our study is to make a choice about the validation method. It is crucial to have a proper and well-researched phase of validation, as it has a direct impact on the results presented. We tried and compared several types of validation methods, ending with the choice of using the last 2 years of data. However, we are fully aware that these short-term previous years might not always be the best representer of the following test year, and this has a clear impact. In addition, one of the two validation years can sometimes be very special (e.g. with very high peaks like in 2011/12, as we can see in Figure 1), and averaging its results with the other year might relatively falsify the results for a given hyperparameter set.

Moreover, in our work, we did not validate the models using extensive sets of hyperparameters. Indeed, we only validated for the main ones (e.g. only alpha for GPs), mainly because of the time constraint. Although we tried to make the most out of validation processes and got very encouraging results, we are fully aware that more substantial studies could be done with the different models used.

It is also important to realize that our experiments were based on the metrics related to ILI rate estimation accuracy only. Other metrics, such as peak week and peak intensity, could also be included to expand the dimension of assessment.

### B. Nowcasting

The outcomes of the proposed nowcasting approaches reinforce the need for various performance metrics to distinguish the strengths of different models in time series regression problems such as influenza prediction. Comparing the tested

Random Forest and Gaussian Processes models in particular, it is consistently true across all every testing period that the former achieves marginally less MAE whereas the latter achieves better RMSE. This discrepancy can be reasoned qualitatively by examining their prediction results in Figure 3.

Throughout all seasons, RFR creates predictions which in general are less noisy than that of other models, with minimum "spikes" throughout periods of low influenza rates (around month 5 to month 10 of every year). This is in contrast with the tested GP model, which produces high-variance predictions throughout the entire test years. The stability of the former model thus results in an overall smaller value of absolute error when summed over the entire test year, thus resulting in lower MAE. Conversely, Random Forest produces particularly poor results at the heights of influenza seasons. The model severely overestimated the height of the 2014/15 peak ILI rate and making similar errors in 2016/17, while underestimating the 2015/16 peak. Comparatively, the GP model produces better ILI rate estimates around seasonal peaks, having captured the 2016/17 peak almost exactly while also more closely predicting the peaks of the 2014/15 and 2015/16. Hence it follows that the RMSE, which more heavily weighs large errors, is higher in our RFR model predictions due to these repeated significant errors observed around seasonal peaks. It might be possible, then, that a good nowcast model can be constructed as a combination of these, where a stable model is used for off-season nowcasting and another used exclusively for seasonal peaks.

Our linear model of choice, being the L1-regularised linear Lasso, performed adequately throughout the three tested seasons, with average results comparable to that of Gaussian Processes. This hence reaffirms, as has been shown to be the case by previous studies [2], [3], [7], that Google queries and ILI rates exhibit a relationship that is approximately linear. However, there are two key differences in this study's findings. The first being that previously referenced studies tested the relationship on ILI rates in the US, with data provided by the CDC, which are different from the data utilised in this study. Secondly, a number of studies noted a linear relationship between the logit-transformed Google search query frequency and ILI rates. As discussed previously, we have not done such data transformations in this study, as preliminary testing on the training dataset showed a lack of improvement in result.

As examined previously, MLP was on average the best performing individual model in our setting. In the first and third test season, MLP outperformed every other model on all metrics by a great margin. In the second test season, however, both RFR and SVR outperformed it on every metric with MLP having a definitively better performance only to Lasso. This inconsistency in the performance of MLP means that we do not have a definitively best individual model among this study, which hence provides us with the incentive to construct an ensemble approach with the aim of capturing positive attributes of these modules.

The proposed ensemble approach outperformed its constituent models substantially in most of our test settings and metrics. On average, the ensemble model saw $10\%$ lower error scores compared to MLP, however, it underperformed in RMSE and correlation in the 2014/15 season. This is likely caused by the other models predicting the height of the seasonal peak to be much higher than that of MLP's (and the ground truth), which collectively results in high RMSE in the ensemble. Despite this, the results obtained have led us to believe that a weighted ensemble of multiple statistical models produces a substantially more consistent and accurate ILI nowcast prediction over the individual supervised machine learning models investigated in this study.

### C. Forecasting

The forecasting MLP has a very different structure compared to the nowcasting version. The validation results suggest that MLP with 0 hidden layers achieves the best performance in most of the ILI seasons. In addition, SVR as a complex forecasting model has also been attempted. Although the model has been tuned properly through the same validation process, it achieves significant worse results compared to MLP and ARIMA. This shows that forecasting is simpler and has stronger linearity than nowcasting. It is likely explained by the incorporation of the nowcast results and thus the model does not need to deal with the query frequencies directly. The dimension of input is much smaller and hence a complex model is not needed.

It is also notable that the *adam* optimizer plays an important role in the 0 hidden layers MLP, as a simple linear regression model using a different optimiser has also been tested but achieved poor results.

The ARIMA model validation method showed us that the best order for all 3 test years was (1,1,2), basically meaning that we only use the last two known days to predict 7 or 14 days ahead. The results showed that a model trained like that can work very well, outperforming the MLP for several metrics in different years. However, we can also see its main drawback in 2014 test season, particularly with 14 days ahead forecasts. When the rates are growing rapidly, using only two days leads to big over predictions and a miss in accurately forecasting the peak (as we can see in figure 4.a).

Unlike nowcasting, where an optimal model is identified, the results on forecasts did not conclusively identify any model as the best. ARIMA produces predictions with high correlations but full of spikes and fluctuations, especially in the first test season. On the other hand, the predictions of MLP are smoother and less noisy but possess a worse RMSE and correlation.

## V. Conclusion & Future Work

In this paper, we presented different statistical approaches to nowcast and forecast ILI rates in the UK, using machine learning models applied on Google search query data and historical ILI rates to obtain relatively accurate predictions. We tested our models on three different influenza seasons, in order to ensure that our models as robust as possible, and

capable of accurately predicting influenza 7-14 days ahead, overcoming the 1-week lag of ILI data reported by the HPA.

Our findings show that, while it is evident that certain supervised models have varying advantages at ILI nowcasting at different metrics, the ensemble of our Lasso, Random Forest Regression, Gaussian Processes, Support Vector Regression, and Multilayer Perceptron models produced the overall best results in most of our test settings. On the other hand, although we tested several linear and nonlinear models to predict influenza rates, we found neural networks and autoregression to be the most accurate, both with competing performance.

The majority of this study was focused on the tuning, validation, and testing of the nowcasting and forecasting models themselves. Furthering this approach, we can extend this study by testing other types of models that were not tested within this study, such as mechanistic models which tend to perform better in tracking an epidemic. Alternatively, future research can be conducted in applying our proposed models using other approaches and data sources. Indeed, with more data (e.g. Twitter, Instagram, etc), we could, for example, add a geographical component and track ILI rates using a network approach, by tracing relations between regions. These types of propositions are actually in the centre of attention, as we are in the middle of a global Covid-19 pandemic, with countries starting to use phone applications to track and forecast the epidemic. If we wanted to continue our study on the novel coronavirus or even still on influenza, we would need to update the query selection or find an automatic way to gather keywords for the filtering step.

## REFERENCES


[1] W. H. Organization, "Influenza (Seasonal), World Health Organization." https://www.who.int/news-room/fact-sheets/detail/influenza-(seasonal). [Online; accessed 10-April-2020].

[2] J. Ginsberg, M. H. Mohebbi, R. S. Patel, L. Brammer, M. S. Smolinski, and L. Brilliant, "Detecting influenza epidemics using search engine query data," *Nature*, vol. 457, no. 7232, p. 1012–1014, 2009.

[3] V. Lampos, A. Miller, S. Crossan, and C. Stefansen, "Advances in nowcasting influenza-like illness rates using search query logs," *Scientific Reports*, vol. 5, 08 2015.

[4] T. Preis and H. Moat, "Adaptive nowcasting of influenza outbreaks using google searches," *Royal Society Open Science*, vol. 1, no. 2, p. 140095, 2014.

[5] M. Santillana, A. Nguyen, M. Dredze, M. Paul, and J. Brownstein, "Combining search, social media, and traditional data sources to improve influenza surveillance," *PLoS computational biology*, vol. 11, 08 2015.

[6] Q. Xu, Y. Gel, L. Ramirez, K. Nezafati, Q. Zhang, and K.-L. Tsui, "Forecasting influenza in hong kong with google search queries and statistical model fusion," *PLoS ONE*, vol. 12, 05 2017.

[7] S. M. Yang, S. and S. Kou, "Accurate estimation of influenza epidemics using google search data via argo," *Proceedings of the National Academy of Sciences*, vol. 112, no. 47, pp. 14473–14478, 2015.

[8] D. McIver and J. Brownstein, "Wikipedia usage estimates prevalence of influenza-like illness in the united states in near real-time," *PLoS Computational Biology*, vol. 10, no. 4, p. 1003581, 2014.

[9] H. Achrekar, A. Gandhe, R. Lazarus, S.-H. Yu, and B. Liu, "Predicting flu trends using twitter data," pp. 702 – 707, 05 2011.

[10] V. Lampos and N. Cristianini, "Tracking the flu pandemic by monitoring the social web," *IAPR 2nd Workshop on Cognitive Information Processing*, 2010.

[11] M. Paul, M. Dredze, and D. Broniatowski, "Twitter improves influenza forecasting," *PLoS currents*, vol. 6, 10 2014.

[12] A. Signorini, A. Segre, and P. Polgreen, "The use of twitter to track levels of disease activity and public concern in the u.s. during the influenza a h1n1 pandemic," *PloS one*, vol. 6, p. e19467, 05 2011.

[13] M. Santillana, A. Nguyen, M. Dredze, M. Paul, and J. Brownstein, "Combining search, social media, and traditional data sources to improve influenza surveillance," *PLoS computational biology*, vol. 11, 08 2015.

[14] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Berlin, Heidelberg: Springer-Verlag, 2006.

[15] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An Introduction to Statistical Learning: With Applications in R*. Springer Publishing Company, Incorporated, 2014.

[16] S. Prabhakaran, "ARIMA Model – Complete Guide to Time Series Forecasting in Python." https://www.machinelearningplus.com/time-series/arima-model-time-series-forecasting-python/. [Online; accessed 21-February-2020].