Automated Machine Learning for wind farms location

Olivier Parisot, Thomas Tamisier
Luxembourg Institute of Science and Technology (LIST)
5 Avenue des Hauts-Fourneaux
4362 Esch-sur-Alzette - Luxembourg
olivier.parisot@list.lu

Keywords:
Automated Machine Learning, wind farms location.

Abstract:
Automated Machine Learning aims at preparing effective Machine Learning models with little or no data science expertise. Tedious tasks like preprocessing, algorithm selection and hyper-parameters optimization are then automatized: end-users just have to apply and deploy the model that best suits the real world problem. In this paper, we experiment Automated Machine Learning to leverage open data sources for predicting potential next wind farms location in Luxembourg, France, Belgium and Germany.

1 INTRODUCTION

The growing application of Machine Learning in a wide range of fields has led to the design of platforms and frameworks facilitating the production of readily actionable models. Even if statistical and coding expertise are still required for data science (Mikalef et al., 2018), such automatized systems are intended to non-experts in classical situations.

In this context, Automated Machine Learning consists in generating and deploying Machine Learning models from an input raw dataset with little or no configuration and coding effort (Hutter et al., 2019). Let us consider the traditional pipeline for supervised classifications, regressions and forecasting tasks:

- Data preprocessing is required to adjust the raw data to the specificity of Machine Learning algorithms (Parisot and Tamisier, 2015): cleansing (missing data imputation or outliers), dimensionality alteration (features removal/creation) and quantity alteration (data sampling or outliers removal).
- A Machine Learning algorithm is then applied to a train model from the preprocessed data.
- Depending on the algorithm, some hyper-parameters have to be optimized to improve the accuracy of the model; in general, this is realized through heuristics requiring heavy computation (Feurer and Hutter, 2019).
- The accuracy of the obtained model is evaluated by computing standard statistical tests (AUC, Precision, Recall, F1, etc.) with a given strategy (holdout or cross-validation).

In practice, all those steps are time-consuming and exposed to methodological errors. Automated Machine Learning platforms aims at systematizing the whole pipeline in order to launch it a number of times with various combinations: several pipelines are tested, the leading models are then compared and the *most accurate* model is finally selected (Raschka, 2018).

In this work, we show how we can apply Automated Machine Learning to estimate the potential location of next wind farms in Luxembourg, France, Belgium and Germany by analyzing data from various available sources.

The rest of this paper is organized as follows. Firstly, existing Automated Machine Learning platforms and their application potential are briefly presented (Section 2). Secondly, the wind farm use case is detailed (Section 3). Finally, the results of experiments with a selection of Automated Machine Learning platforms are discussed (Section 4).

2 RELATED WORKS

Numerous Automated Machine Learning systems were developed in recent years and an exhaustive list was already compiled (Milutinovic et al., 2020). We can consider two kinds of solutions:

- Open source tools like TPOT (Olson and Moore, 2016), Auto-Weka (Kotthoff et al., 2017) and Auto-sklearn (Feurer et al., 2019): these frameworks are mainly based on existing data mining tools (sklearn, Weka) and were recently benchmarked (Gijsbers et al., 2019).
- Commercial cloud-based solutions such as IBM Watson AutoAI, Google Cloud AutoML, Microsoft Azure AutoML: these platforms are integrated in complete Machine Learning development environments and managed through a graphical user interface.

As regard the application potential, Automated Machine Learning is more and more applied in various real world problems. Below are some recent examples:

- Logistic: optimization of commute times at Toronto by analysing a database with past transport delays (Cao et al., 2019).
- Education: support to educators in anticipating students progression in online learning platforms (Tsiakmaki et al., 2020).
- Healthcare: improvement of the care quality for British patients by analysing biomedical databasets (Waring et al., 2020).

Moreover, recent works concern trustability of models obtained with Automated Machine Learning. As an example, a study has investigated why experienced data scientists would use such automatic approaches. An other article even produces human-readable reports with natural language to interpret automatically obtained models (Steinruecken et al., 2019).

In this paper, we apply Automated Machine Learning to produce a model to assess the suitability of a given area to host wind farms. Even if various computational methods for wind farms placement have been proposed (Shahab and Singh, 2019), no specific approach based on Automated Machine Learning seems available.

3 USE CASE

Nowadays, the production of wind energy is more than ever an environmental and political priority (Hevia-Koch and Ladenburg, 2019). We can notice the installation of new wind farms everywhere. To understand and therefore anticipate this trend from a data-driven approach, Machine Learning can be useful. Positioning of wind farms is a complex problem: weather conditions like wind speed are obviously important to justify their location, but other parameters such as environment, performance, societal impact are taken into account (Kazak et al., 2017).

To collect data that could be meaningful for the prediction of the next wind farms locations in Luxembourg, France, Belgium and Germany, we have considered various open data sources:

- Geographical locations of the current onshore wind farms [1].
- Historical time series of daily minimal/maximal/average wind speed values: each time serie corresponds to a geolocated area and two years of data have been considered.
- Elevations of the current wind farms by using the STRM digital elevation model [2]: it may help determine how much topology is taken into account for installing wind farms.
- Cities positions and populations [3]: it may help to check the relationships between wind farms locations and town centers.
- Points of Interests positions [4]: it may have an impact on wind farms installation.

Combining these heterogeneous data sources and by considering geographical areas with a width of 2500 meters, we have built an aggregated dataset with ten features (Table 1). The last feature is a binary flag indicating if the location holds at least a wind farm and can be used as a decision class. The area width was empirically defined after several experiments and by considering the weather data coverage.

As a result, we considered the prediction of next wind farms location as a supervised binary classification problem. The input dataset is class-imbalanced since most of the considered geographical areas do not accommodate wind farms. The challenge of dealing with such imbalanced classes is found in various domains like medical diagnosis or fraud detection (Haixiang et al., 2017).

In the next section, we detail some experiments to check if an accurate model can be obtained with a selection of Automated Machine Learning platforms.

[1]https://data.open-power-system-data.org
[2]http://srtm.csi.cgiar.org
[3]https://www.geonames.org/
[4]http://openpoimap.org

Table 1: The structure of the dataset describing the location of existing onshore wind farms in Luxembourg/France/Belgium/West of Germany (117278 records).

Numeric	Latitude of the geographical area (degrees)
Numeric	Longitude of the geographical area (degrees)
Numeric	Elevation of the geographical area (meters)
Numeric	Average wind speed on a recent time frame (meters per second)
Numeric	Maximum wind speed on a recent time frame (meters per second)
Numeric	Distance between the geographical area center and the nearby POI (meters)
Numeric	Count of POIs in the considered geographical area
Numeric	Distance between the geographical area center and the nearby town (meters)
Numeric	Population of the nearby town
Binary	TRUE if there is at least a wind farm in the geographical area, FALSE otherwise

4 EXPERIMENTS

Starting from the previously described input dataset, we have tested two commercial cloud platforms (IBM Watson AutoAI, Microsoft Azure AutoML) and an open source tool (TPOT). The goal is to build an efficient Machine Learning model for wind farms prediction rather than realizing a strict evaluations of Automated Machine Learning platforms; it was already proposed by (Milutinovic et al., 2020).

The next sections detail the results obtained with these different approaches (Table 2). Each statistical test is computed by following the hold-out strategy (Kohavi et al., 1995): 90% for training set, 10% for test set.

4.1 Simple Python code

To have a point of reference, we have developed a simple Python source code to run the standard Gradient Tree Boosting algorithm with the widely-used scikit-learn package (Pedregosa et al., 2011). The script executes the following steps: data loading, no data preprocessing, training with default parameters and then evaluation of the leading model.

```
...
rfc=GradientBoostingClassifier()
trainingF =[...]
trainingT =[...]
rfc.fit(trainingF ,trainingT)
...
```

As a result, the output model has a good-enough Precision score but a insufficient Recall score (Table 2). The next sections will show if Automated Machine Learning platforms can provide better results for the current use case.

4.2 IBM Watson AutoAI

Watson AutoAI is a module integrated into the online IBM Watson Machine Learning service which is mainly accessible through a graphical user-interface (Figure 1).

The result of our test with Watson AutoAI is as follows: after the data ingestion and some computation time, the platform proposes a short list of four predictive models obtained with the LGBM classifier (i.e. an implementation of Gradient Boosting Tree). The best model is obtained with the following pipeline:

- The features engineering phase leads to the generation of twelve additional features like *square(sin(elevation))*.
- Two hyper-parameters have been optimized though optimizations are not clarified.

Despite various efforts to optimize the configuration, we did not succeed in improving the best model produced by of Watson AutoML, above the score (low F1: 0.343). The model obtained with the simple Python script remains therefore the best one.

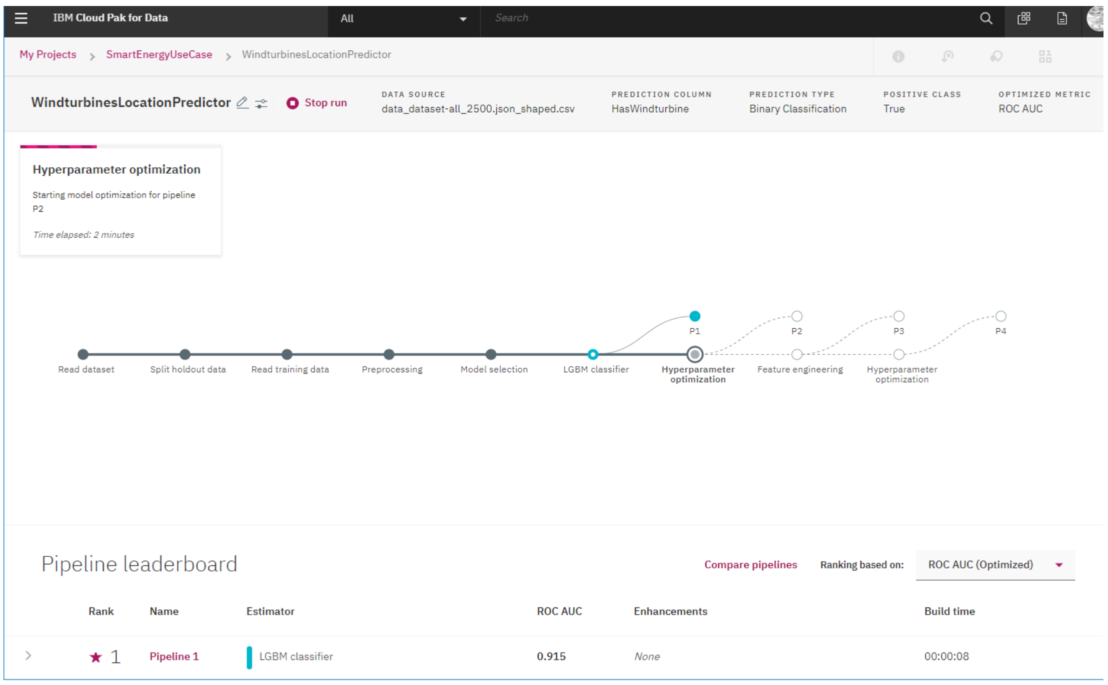
4.3 Microsoft Azure AutoML

AutoML is integrated in Azure Machine Learning Studio, an integrated development environment for designing Machine Learning workflow on the Microsoft cloud platform.

From a user point of view, Azure AutoML and IBM Watson AutoAI are slightly different. Once input data are loaded into Microsoft Azure AutoML, a *data guards* step checks the data characteristics and produces warnings if some issues are detected (missing data, imbalanced class, cardinaltity check, etc.). These warnings do not trigger automatical data preprocessing: they are just provided to inform the end-user that those issues

Table 2: Hold-out evaluation of the produced models for wind farms location prediction (90% for training set, 10% for test set). Two commercial platforms were tested, an open-source solution was used and a simple Python source code was implemented to have a reference.

Test	Preprocessing	Best algorithm	Precision	Recall	F1
Python code	Nothing	Gradient Tree Boosting	0.825	0.540	0.567
Watson AutoAI	Described	LGBM	0.240	0.599	0.343
Azure AutoML	Not described	VotingEnsemble	0.783	0.623	0.670
TPOT	Fully reproductible	Random Forests cascade	0.759	0.693	0.721



WindturbinesLocationPredictor

DATA SOURCE: data_dataset-all_2500.json_shaped.csv
PREDICTION COLUMN: HasWindturbine
PREDICTION TYPE: Binary Classification
POSITIVE CLASS: True
OPTIMIZED METRIC: ROC AUC

Hyperparameter optimization

Starting model optimization for pipeline P2

Time elapsed: 2 minutes

Pipeline leaderboard

Ranking based on: ROC AUC (Optimized)

Rank	Name	Estimator	ROC AUC	Enhancements	Build time
★ 1	Pipeline 1	LGBM classifier	0.915	None	00:00:08

Figure 1: Interactive Dashboard of IBM Watson Studio AutoAI: the Automated Machine Learning steps are then represented during the computation of a predictive model for next wind farms location.

may have an impact on the final results. After these checks, Azure AutoAI launches a list of 74 jobs to run various pipelines (data preprocessing, algorithm selection, hyper-parameters optimization). At the end, the best model has good estimators (F1, Precision, Recall) that are really similar to the one obtained with Python source code. This model is computed with this pipeline:

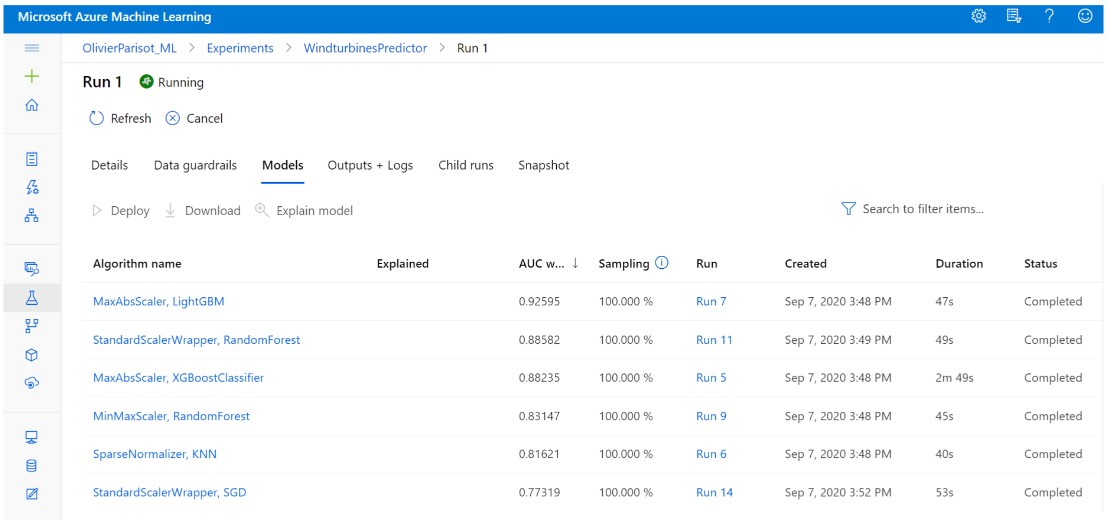
- Algorithm: Voting Ensemble, i.e. a weighted ensemble of other models (not described).
- Features preprocessing and hyper-parameters optimization are not described too.

4.4 TPOT

TPOT (Tree-based Pipeline Optimization Tool) is an open source Automated Machine Learning solution based on a Genetic Algorithm (Olson and Moore, 2016) As a Python package, TPOT aims at building optimized classifications and regressions models by writing some source code or by launching a configurable command-line tool. In this work, we have used the second method to analyze our dataset (Figure 3).

The resulting model is much better than those obtained with the Python script and with the tested Automated Machine Learning platforms. The following piece of Python source code is generated by TPOT and allows to understand exactly the pipeline leading to the best model (data preprocessing, algorithm training, hyper-parameters optimization):

```
...
pipeline=make_pipeline(
StackingEstimator(
estimator=RandomForestClassifier(
  bootstrap=False,
  criterion="entropy",
  max_features=0.55,
```


Microsoft Azure Machine Learning

OlivierParisot_ML > Experiments > WindturbinesPredictor > Run 1

Run 1 Running

Refresh Cancel

Details Data guardrails Models Outputs + Logs Child runs Snapshot

Deploy Download Explain model Search to filter items...

Algorithm name	Explained	AUC w... ↓	Sampling	Run	Created	Duration	Status
MaxAbsScaler, LightGBM		0.92595	100.000 %	Run 7	Sep 7, 2020 3:48 PM	47s	Completed
StandardScalerWrapper, RandomForest		0.88582	100.000 %	Run 11	Sep 7, 2020 3:49 PM	49s	Completed
MaxAbsScaler, XGBoostClassifier		0.88235	100.000 %	Run 5	Sep 7, 2020 3:48 PM	2m 49s	Completed
MinMaxScaler, RandomForest		0.83147	100.000 %	Run 9	Sep 7, 2020 3:48 PM	45s	Completed
SparseNormalizer, KNN		0.81621	100.000 %	Run 6	Sep 7, 2020 3:48 PM	40s	Completed
StandardScalerWrapper, SGD		0.77319	100.000 %	Run 14	Sep 7, 2020 3:52 PM	53s	Completed

Figure 2: Interactive Dashboard of Microsoft Azure AutoML during the generation of a machine learning model for next wind farms location prediction.

```
  min_samples_leaf=3,
  min_samples_split=16)),
RandomForestClassifier(
  bootstrap=False,
  criterion="entropy",
  max_features=0.65,
  min_samples_leaf=10,
  min_samples_split=16)
)
trainingF =[...]
trainingT =[...]
pipeline.fit(trainingF,trainingT)
...
```

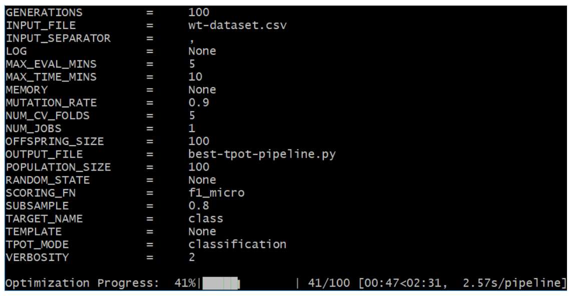
4.5 Discussion

According to our experiments (Table 2), we have observed that the tested Automated Machine Learning platforms can produce different pipelines and then models for the prediction of next wind farms location. Even if the computation takes time and resources, the accuracy of the yielded out models does not always supasses the accuracy of the simple Python source code.

Azure AutoAI provides a good model with few effort (no source code and very little configuration) while Watson AutoML generates a poor one: it could be explained by the class-imabalanced input dataset (Azure AutoAI has detected this point, AutoML not). By using these two commercial platforms, the resulting models can be deployed and then used with one click. However, we notice some limitations:

- The first one is due to the feature engineering phase: it may lead to features creation that are hard to interpret (example: *square(sin(elevation))* in Watson AutoML), or the result is not described at all by the platform (Azure AutoML). As shown in (Drozdal et al., 2020), it may be a major drawback in case of we need to adapt the model.
- A further limitation is the lack of customisation in order to adapt manually the resulting pipelines (preprocessing and hyperparameters). There is now way to change anything, the end-user can just choose a model among the list of those which were computed.

According to our experiments, an open source tool like TPOT produces better models to predict the next wind farms location, It is clearly more complex to use for non experts (there is no user-friendly interface and the deployment requires a lot of coding), but it allows a higher level of customization (many parameters can be set before the Automated Machine Learning process and it


```
GENERATIONS          =     100
INPUT_FILE           =     wt-dataset.csv
INPUT_SEPARATOR      =     ,
LOG                  =     None
MAX_EVAL_MINS        =     5
MAX_TIME_MINS        =     10
MEMORY               =     None
MUTATION_RATE        =     0.9
NUM_CV_FOLDS         =     5
NUM_JOBS             =     1
OFFSPRING_SIZE       =     100
OUTPUT_FILE          =     best-tpot-pipeline.py
POPULATION_SIZE      =     100
RANDOM_STATE         =     None
SCORING_FN           =     f1_micro
SUBSAMPLE            =     0.8
TARGET_NAME          =     class
TEMPLATE             =     None
TPOT_MODE            =     classification
VERBOSITY            =     2

Optimization Progress:  41%|████       | 41/100 [00:47<02:31,  2.57s/pipeline]
```

Figure 3: Command-line Dashboard of TPOT during the computation of an optimized model.

is possible to generate ready-to-use Python code for running computed models).

5 CONCLUSION

In this work, we applied Automated Machine Learning to build a model checking if a geographical zone is suitable to host wind farms in Luxembourg, France, Belgium and Germany. A relevant dataset was built from open data sources and predictive models have been trained with various commercial and open source Automated Machine Learning platforms.

In future work, we will take advantage of High-Performance Computing infrastructure (HPC) to efficiently analyze the evolution over time of wind farms installation policies.

Acknowledgments: This work was funded by the FEDER Data Analytics Platform project (http://tiny.cc/feder-dap-project). Special thanks to Anne Hendrick, Samuel Renault and Raynald Jadoul for their support.

REFERENCES

Cao, T., Roy, D., and Nedelescu, T. (2019). Optimizing Commute Time with IBM Watson Studio. In *CCSE 2019*, pages 388–390.

Drozdal, J., Weisz, J., Wang, D., Dass, G., Yao, B., Zhao, C., Muller, M., Ju, L., and Su, H. (2020). Trust in AutoML: exploring information needs for establishing trust in automated machine learning systems. In *ACM IUI 2020*, pages 297–307.

Feurer, M. and Hutter, F. (2019). Hyperparameter optimization. In *Automated Machine Learning*, pages 3–33. Springer, Cham.

Feurer, M., Klein, A., Eggensperger, K., Springenberg, J. T., Blum, M., and Hutter, F. (2019). Auto-sklearn: efficient and robust automated machine learning. In *Automated Machine Learning*, pages 113–134. Springer, Cham.

Gijsbers, P., LeDell, E., Thomas, J., Poirier, S., Bischl, B., and Vanschoren, J. (2019). An open source AutoML benchmark. *arXiv preprint*.

Haixiang, G., Yijing, L., Shang, J., Mingyun, G., Yuanyue, H., and Bing, G. (2017). Learning from class-imbalanced data: Review of methods and applications. *Expert Systems with Applications*, 73:220–239.

Hevia-Koch, P. and Ladenburg, J. (2019). Where should wind energy be located? a review of preferences and visualisation approaches for wind turbine locations. *Energy Research and Social Science*, 53:23–33.

Hutter, F., Kotthoff, L., and Vanschoren, J. (2019). *Automated machine learning: methods, systems, challenges*. Springer Nature.

Kazak, J., Van Hoof, J., and Szewranski, S. (2017). Challenges in the wind turbines location process in central europe–the use of spatial decision support systems. *Renewable and Sustainable Energy Reviews*, 76:425–433.

Kohavi, R. et al. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Ijcai*, volume 14, pages 1137–1145. Montreal, Canada.

Kotthoff, L., Thornton, C., Hoos, H. H., Hutter, F., and Leyton-Brown, K. (2017). Auto-WEKA 2.0: Automatic model selection and hyperparameter optimization in WEKA. *The Journal of Machine Learning Research*, 18(1):826–830.

Mikalef, P., Giannakos, M. N., Pappas, I. O., and Krogstie, J. (2018). The human side of big data: Understanding the skills of the data scientist in education and industry. In *IEEE Global Engineering Education Conference*, pages 503–512.

Milutinovic, M., Schoenfeld, B., Martinez-Garcia, D., Ray, S., Shah, S., and Yan, D. (2020). On evaluation of AutoML systems.

Olson, R. S. and Moore, J. H. (2016). TPOT: A tree-based pipeline optimization tool for automating machine learning. In *Workshop on automatic machine learning*, pages 66–74. PMLR.

Parisot, O. and Tamisier, T. (2015). An interactive tool for transparent data preprocessing. *ERCIM NEWS*, 100(1-4):33.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al. (2011). Scikit-learn: Machine learning in python. *Journal of ML research*, 12:2825–2830.

Raschka, S. (2018). Model evaluation, model selection, and algorithm selection in machine learning. *arXiv preprint*.

Shahab, A. and Singh, M. (2019). Comparative analysis of different machine learning algorithms in classification of suitability of renewable energy resource. In *ICCSP 2019*, pages 0360–0364.

Steinruecken, C., Smith, E., Janz, D., Lloyd, J., and Ghahramani, Z. (2019). The automatic statistician. In *Automated Machine Learning*, pages 161–173. Springer, Cham.

Tsiakmaki, M., Kostopoulos, G., Kotsiantis, S., and Ragos, O. (2020). Implementing AutoML in educational data mining for prediction tasks. *Applied Sciences*, 10(1):90.

Waring, J., Lindvall, C., and Umeton, R. (2020). Automated machine learning: Review of the state-of-the-art and opportunities for healthcare. *Artificial Intelligence in Medicine*, page 101822.