

УДК 621.791.722
DOI: 10.15827/0236-235X.134.316-323

Дата подачи статьи: 22.10.20
2021. Т. 34. № 2. С. 316–323

Модель анализа и прогнозирования технологических параметров для процесса электронно-лучевой сварки

В.С. Тынченко¹, к.т.н., доцент, vadimond@mail.ru
С.О. Курашкин¹, аспирант, scorpion_ser@mail.ru
И.А. Головенко¹, магистрант, golovonokia@mail.ru
В.Е. Петренко¹, аспирант, dpbra@inbox.ru

¹ Сибирский государственный университет науки и технологий
им. академика М.Ф. Решетнева, кафедра информационно-управляющих систем,
г. Красноярск, 660037, Россия

Целью исследования являются создание математической модели анализа и прогнозирования технологических параметров для процесса электронно-лучевой сварки с помощью современных регрессионных моделей, а также ее реализация в виде программной системы на языке программирования Python с применением программных пакетов Scikit-learn, Pandas, NumPy и Matplotlib.

В сущности задача предсказания параметров технологического процесса электронно-лучевой сварки представляет собой задачу регрессии, для решения которой существует множество подходящих алгоритмов. В рамках данной работы используются алгоритм регрессионного анализа как полиномиальная регрессия с $L2$ регуляризацией – гребневая регрессия, а также ансамбль алгоритмов решающих деревьев – случайный лес. Использование разработанной предсказательной модели позволит технологу более осознанно подходить как к выбору диапазона варьируемых параметров для исследований в новых технологических режимах, так и к поднятию качества в уже отработанных технологических режимах. Применение предложенных методов также позволит снизить временные и трудовые затраты на поиск, отработку и наладку технологического процесса.

В работе описывается алгоритм гребневой регрессии, анализируется применимость данного алгоритма к решению поставленной задачи, а также проверяется достоверность прогнозов, получаемых при их непосредственном использовании. Кроме того, рассматривается процесс обучения модели на основе данных, полученных в рамках проведения экспериментов по отработке технологического процесса электронно-лучевой сварки.

Анализ показал, что допустимо использование предложенного метода для технологических процессов, имеющих подобные статистические зависимости. Внедрение предложенного подхода предсказания параметров электронно-лучевой сварки на производстве позволит осуществить поддержку принятия технологических решений при отработке технологического процесса электронно-лучевой сварки, а также при вводе в производство новых видов продукции.

Ключевые слова: электронно-лучевая сварка, технологический процесс, полиномиальная регрессия, ансамбль алгоритмов, машинное обучение, электронный пучок, программный продукт, Python, Scikit-learn, Pandas.

Технологические процессы получения неразъемных соединений различных изделий аэрокосмического назначения с использованием технологий электронно-лучевой сварки основаны на повторяемости отработанного ранее режима предварительной подготовки и непосредственной работы оборудования [1–3].

В числе самых важных этапов технологического процесса – наладка и отработка заданных режимов, по завершении которых можно судить о повторяемости результатов и степени надежности конечных изделий [4–6].

Для технологических процессов остро стоит вопрос подбора или осознанного выбора оптимальных параметров, которые зависят от

предъявляемых к конечному изделию критериев качества [7–9]. Более того, данное утверждение справедливо и для способов поиска улучшения или преобразования уже налаженного технологического процесса, например, в ситуации, когда необходимо улучшить один из параметров, определяющих качество конечного изделия, не изменяя остальные или же не допуская их отклонения на определенную величину [10–12]. Однако такие процессы, как электронно-лучевая сварка, относительно сложно поддаются наладке или изменению в связи либо с недостаточной изученностью, либо с комплексной сложностью, когда невозможно учесть все факторы в том виде, который

позволил бы однозначно определять потенциальные изменения и влияние параметров на процесс в целом [13–15]. К тому же при необходимости перехода к другому типу соединений или улучшения качества соединения по некоторым из оцениваемых параметров требуется проведение как наладки и отработки, так и поисковых исследований такого режима, при котором достигаются заданные критерии качества конечного изделия [16–18]. Данные этапы влекут за собой большие накладные расходы, в частности, занятие ресурса оборудования, специалистов, планирующих, а также непосредственно проводящих эксперименты и обрабатывающих их результаты в рамках изменяемого или формируемого технологического процесса. Кроме того, каждый отдельный эксперимент приводит к заведомо уничтоженному образцу изделия, которое в дальнейшем подвергается лабораторным исследованиям на соответствие критериям качества для неразъемного соединения. Во всей этой цепочке нужно получить оптимум, чтобы уменьшить как временные, так и экономические затраты [19–21].

Все вышесказанное обуславливает необходимость поиска методов упрощения процессов наладки и преобразования технологических процессов.

В данной работе рассмотрены и реализованы модели машинного обучения для задач регрессии, благодаря которым становится возможным построение аппроксимированных зависимостей, позволяющих технологам более осознанно подходить к вопросам как выбора диапазона варьируемых параметров для исследований в новых технологических режимах, так и поднятия качества в уже отработанных.

Описание применяемых моделей

Модель Ridge. Построение простой линейной модели с коэффициентами $w = (w_1, \dots, w_p)$ в пакете Scikit-learn происходит через минимизацию остаточной суммы квадратов между X и y : $\min_w \|Xw - y\|_2^2$.

В Scikit-learn реализованы линейные модели с $L2$ и $L1$ регуляризацией – Ridge, Lasso.

Ridge решает проблемы обычных наименьших квадратов путем наложения штрафа на размер коэффициентов. Коэффициенты гребня минимизируют штрафную остаточную сумму квадратов: $\min_w \|Xw - y\|_2^2 + \alpha \|w\|_2^2$.

Параметр сложности (alpha) $\alpha \geq 0$ контролирует величину усадки: чем больше значение α ,

тем больше величина усадки и, следовательно, коэффициенты становятся более устойчивыми к коллинеарности.

Lasso представляет собой линейную модель, оценивающую разреженные коэффициенты. Это полезно в некоторых контекстах из-за тенденции отдавать предпочтение решениям с меньшим количеством ненулевых коэффициентов, эффективно уменьшая количество параметров, от которых зависит данное решение [11–13].

Для создания полиномиальной регрессии воспользуемся предварительной обработкой из Scikit-learn – Polynomial Features. Она создает новую матрицу объектов, состоящую из всех полиномиальных комбинаций объектов со степенью (degree), меньшей или равной указанной. После расширения пространства гипотез до всех полиномов степени p (degree = p) линейная модель будет иметь вид

$$f(X) = w_0 + w_1x_1 + w_2x_2 + \dots + w_px_p + w_{12}x_1^2 + \dots + w_{22}x_1x_2 + \dots + w_{p2}x_1x_p + \dots + w_{pp}x_p^2 + \dots$$

Если degree = 2, линейная модель для четырех параметров примет вид

$$f(X) = w_0 + w_1x_1 + w_2x_2 + w_3x_3 + w_4x_4 + w_{11}x_1^2 + w_{12}x_1x_2 + w_{13}x_1x_3 + w_{14}x_1x_4 + w_{22}x_2^2 + w_{23}x_2x_3 + w_{24}x_2x_4 + w_{33}x_3^2 + w_{34}x_3x_4 + w_{44}x_4^2.$$

Набор данных будет стандартизирован с помощью обработки – Standard Scaler. Стандартизация набора данных является общим требованием для многих оценщиков машинного обучения: они могут показывать плохие результаты, если отдельные параметры не выглядят более или менее похожими на стандартные нормально распределенные данные [12–14].

Стандартизация набора данных в Standard Scaler происходит путем удаления среднего значения и масштабирования до дисперсии единиц.

Стандартная оценка для каждой переменной h из обучающей выборки рассчитывается

$$\text{как } z = \frac{(h - u)}{s}, \text{ где } u - \text{среднее значение обучающих выборок; } s - \text{стандартное отклонение обучающих выборок.}$$

Основные гиперпараметры в модели Ridge, которые подбираются для поиска оптимального решения, это degree – степень полинома и alpha – сила регуляризации.

Модель Random Forest Regressor. Реализацией модели «случайный лес» в пакете Scikit-

learn является Random Forest Regressor (RFR). RFR включает в себя алгоритм усреднения, основанный на рандомизированных деревьях решений, – метод Extra-Trees. Алгоритм представляет собой методы возмущения и объединения, специально разработанные для деревьев.

В RFR каждое дерево в ансамбле строится из выборки, взятой с заменой (метод bootstrap) из обучающего набора. Кроме того, при разделении каждого узла во время построения дерева наилучшее разделение определяется по всем входным параметрам.

Алгоритм построения случайного леса, состоящего из N (`n_estimators`) деревьев, выглядит следующим образом.

Для каждого $n = 1, \dots, N$

1) генерируется выборка X_n с помощью инструментария bootstrap;

2) происходит построение решающего дерева b_n по выборке X_n :

- по заданному критерию (`criterion`) выбирается лучший признак, делается разбиение в дереве по нему, и так до исчерпания выборки;

- дерево строится, пока в каждом листе не более n_{\min} объектов (`min_samples_leaf`) или пока не будет достигнута определенная глубина дерева (`max_depth`);

- при каждом разбиении сначала выбираются m случайных признаков (`max_features`) из n исходных, и оптимальное разделение выборки ищется только среди них.

Итоговое решение является средним значением: $a(x) = \frac{1}{N} \sum_{i=1}^N b_i(x)$.

Основные гиперпараметры в RFR, которые подбирались для поиска оптимального решения:

- `n_estimators` – число деревьев в лесу;
- `criterion` – функция, измеряющая качество разбиения ветки дерева (MSE, MAE);
- `max_depth` – максимальная глубина дерева;
- `max_features` – число признаков, по которым ищется разбиение;
- `min_samples_leaf` – минимальное число объектов в листе;
- `min_samples_split` – минимальное количество объектов, необходимое для разделения внутреннего узла дерева.

Модель Gradient Boosting Regressor. В рассматриваемом случае градиентный бустинг будет над решающими деревьями. Он строит модель предсказания в форме ансамбля слабых

предсказывающих моделей деревьев решений. В Scikit-learn модель Gradient Boosting Regressor (GBR) строит аддитивную модель поэтапно, что позволяет оптимизировать произвольные дифференцируемые функции потерь. На каждом этапе дерево решений соответствует отрицательному градиенту заданной функции потерь.

Опишем алгоритм GBR.

1. Инициализируется модель константным значением $\hat{f}(x) = f_0$, $f_0 = \gamma$, $\gamma \in \mathbb{R}$:

$$f_0 = \arg \min_{\gamma} \sum_{i=1}^n L(y_i, \gamma).$$

2. Для каждой итерации $t = 1, \dots, M$ ($M = n_estimators$) повторяются

- подсчет псевдоостатков r_i :

$$r_{it} = - \left[\frac{\partial L(y_i, f(x_i))}{\partial f(x_i)} \right]_{f(x)=\hat{f}(x)}, \text{ for } i = 1, \dots, n;$$

- построение нового базового алгоритма $h_t(x)$ как регрессии на псевдоостатках $\{(x_i, r_{it})\}_{i=1, \dots, n}$;

- нахождение оптимального коэффициента ρ_t при $h_t(x)$ относительно исходной функции потерь (loss):

$$\rho_t = \operatorname{argmin}_{\rho} \sum_{i=1}^n L(y_i, \hat{f}(x_i) + \rho \cdot h(x_i, \theta));$$

- сохранение: $f_t(x) = \rho_t \cdot h_t(x)$;

- обновление текущего приближения:

$$\hat{f}(x) \leftarrow \hat{f}(x) + f_t(x) = \sum_{i=0}^t f_i(x).$$

3. Компоновка итоговой модели:

$$\hat{f}(x) = \sum_{i=0}^M f_i(x).$$

Основные гиперпараметры в GBR, которые подбирались для поиска оптимального решения:

- `n_estimators` – количество этапов повышения градиента (количество используемых слабых деревьев решений);
- `loss` – функция потерь для оптимизации (MSE, MAE);
- `max_depth` – максимальная глубина каждого дерева решений;
- `max_features` – число признаков, по которым ищется разбиение;
- `min_samples_split` – минимальное количество объектов, необходимое для разделения внутреннего узла дерева;
- `min_samples_leaf` – минимальное число объектов в листе.

Для подбора оптимальных гиперпараметров в моделях применялась обработка в Scikit-learn Grid Search CV – исчерпывающий поиск по указанным значениям гиперпараметров для модели. Он реализует метод подбора и оценки. Гиперпараметры модели оптимизируются путем перекрестной проверки (cross-validation) по сетке гиперпараметров.

Главные параметры Grid Search CV:

- estimator – модель, в которой проходит подбор;
- param_grid – наборы гиперпараметров, которые необходимо проверить;
- scoring – метрика, по которой будет происходить оценивание;
- cv – количество блоков в перекрестной проверке.

Реализация моделей в Python 3

Для реализации программы применен язык программирования Python version 3.8 [15–17].

В Python 3, помимо стандартных, использовались перечисленные далее библиотеки.

Библиотека Scikit-learn version 0.22.2, реализующая машинное обучение на Python, с простыми и эффективными инструментами для прогнозирования анализа данных [18, 19].

Pandas version 1.0.3 – мощный инструмент для анализа и обработки данных [19, 20].

NumPy version 1.18.2 – фундаментальный пакет для научных вычислений на Python [19].

Matplotlib version 3.2.1 – библиотека черчения для языка программирования Python и его числового математического расширения NumPy [21].

Tkinter – пакет графического пользовательского интерфейса Python [18–20].

Реализации моделей для задач регрессии в Python 3 будут иметь следующие названия: Ridge – гребневая регрессия (полиномиальная, множественная регрессия с $L2$ регуляризацией), Random Forest Regressor – ансамбль «случайный лес» для задач регрессии, Gradient Boosting Regressor – градиентный бустинг для задач регрессии.

Проектирование программной системы

Система представляет собой экспериментальный аналитический блок для анализа и прогнозирования результатов технологического процесса. Она позволяет выбирать базу экспериментов технологического процесса, па-

раметры и результаты процесса, модель прогнозирования, а также анализировать данные, выбирать, настраивать и обучать модель на БД экспериментов. С помощью обученной модели есть возможность визуализировать спрогнозированные ею данные и предсказывать результаты технических процессов.

Возможные варианты использования системы для пользователя представлены на рисунке 1.

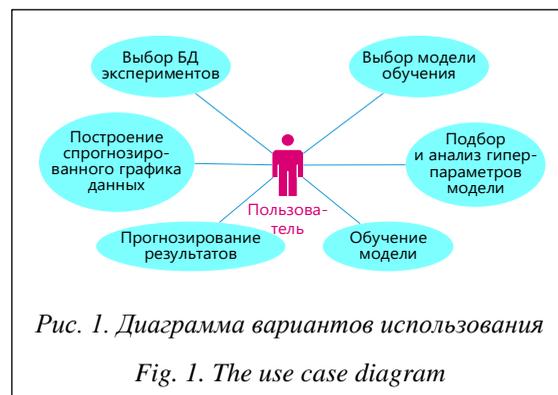


Рис. 1. Диаграмма вариантов использования

Fig. 1. The use case diagram

Система состоит из главного окна и рабочих окон моделей.

Блок-схема работы с системой представлена на рисунке 2.

Как видно из блок-схемы, на начальном этапе происходит выбор базы экспериментов, затем – параметров и результата процесса. Далее проверяется, требуется ли провести анализ данных. В случае необходимости осуществляется анализ данных, на основе которого выбирается модель прогнозирования. В отдельном окне производится визуализация гиперпараметров. Затем происходит обучение модели на основе разработанных алгоритмов, визуализируются полученные данные в ходе обучения и строится прогноз модели.

Описание программной системы

Для успешного выполнения программы необходимо наличие следующего установленного ПО: операционная система Windows 7 и выше, программа Python 3.6 или выше, пакеты для Python – Scikit-learn 0.22.2, Pandas 1.0.3, NumPy 1.18.2, Matplotlib 3.2.1 или выше.

Главное окно системы представлено на рисунке (см. <http://www.swsys.ru/uploaded/image/2021-2/2021-2-dop/23.jpg>).

В главном окне отображается непосредственно загруженная БД. В нем выбираются параметры технического процесса и его ре-

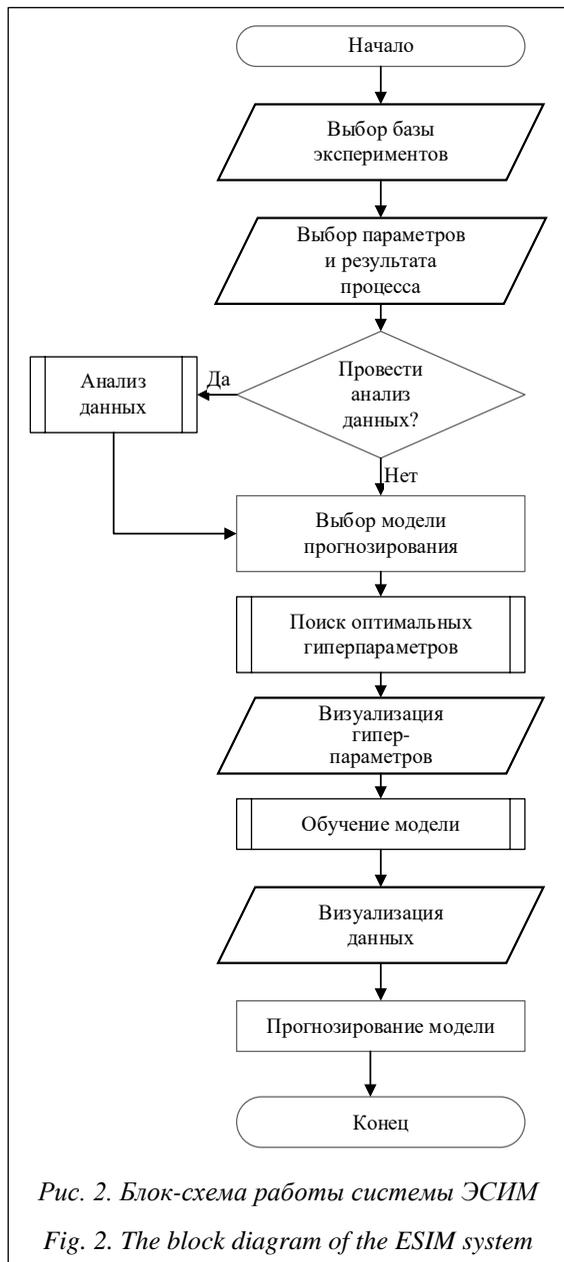


Рис. 2. Блок-схема работы системы ЭСИМ
 Fig. 2. The block diagram of the ESIM system

зультаты. После выбора переменных можно построить корреляционную матрицу.

Вверху главного окна отображаются кнопки моделей обучения Ridge, RFR, GBR.

На рисунке 3 представлено окно модели Ridge. В разделе Grid Search выбирается количество блоков кроссвалидации, при помощи которой подбираются оптимальные гиперпараметры модели. Лучшие результаты поиска отображаются в текстовом поле. Строятся графики изменения гиперпараметров.

В разделе Fit Model происходит обучение модели. Есть возможность оставить подобранные системой гиперпараметры либо ввести свои значения. После обучения модели с вы-

бранными гиперпараметрами отображаются оценки обученной модели.

В разделе Data graph происходит визуализация данных, спрогнозированной моделью, в виде интерактивного графика в отдельном окне (рис. 4). В данном окне в реальном времени можно изменять параметры процесса и наблюдать изменения зависимостей между параметрами и результатами процесса.

В разделе Model prediction указываются значения параметров технического процесса и прогнозируются результаты.

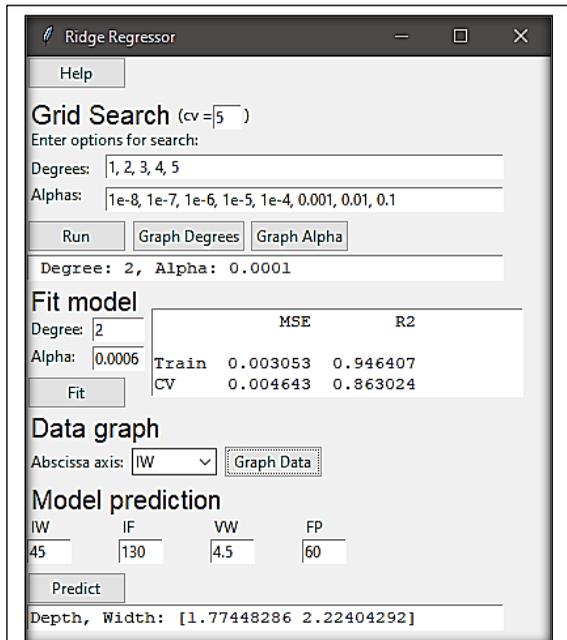


Рис. 3. Окно модели Ridge
 Fig. 3. The Ridge model window

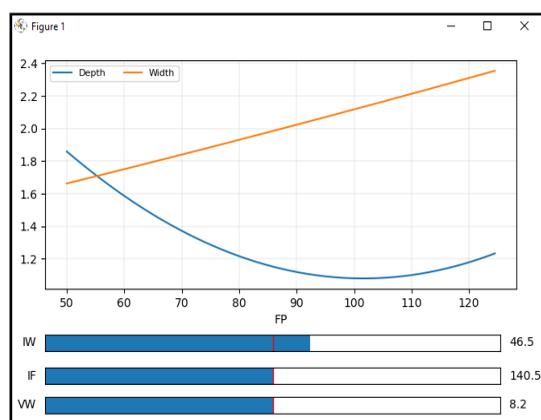


Рис. 4. Пример графика данных, спрогнозированной моделью
 Fig. 4. The data curve example predicted by the model

В окнах моделей RFR и GBR (см. <http://www.swsys.ru/uploaded/image/2021-2/2021-2-dop/21.jpg>) с помощью рандомизированного подбора осуществляется выбор оптимальных гиперпараметров и указывается количество операций в подборе.

Остальные функции и возможности аналогичны модели Ridge.

Заключение

В статье описана разработанная модель прогнозирования для процесса электронно-луче-

вой сварки с наилучшей точностью прогнозирования 93 % – модель Ridge.

Применение разработанного программного модуля предсказания параметров электронно-лучевой сварки на производстве позволит осуществить поддержку принятия технологических решений при отработке технологического процесса электронно-лучевой сварки и при вводе в производство новых видов продукции, а также снизить временные и трудовые затраты на поиск, отработку и наладку технологического процесса.

Исследование выполнено при финансовой поддержке РФФИ, Правительства Красноярского края и Краевого фонда науки в рамках научного проекта № 19-48-240007.

Литература

1. Weglowski M.S., Blacha S., Phillips A. Electron beam welding – Techniques and trends – Review. Vacuum, 2016, vol. 130, pp. 72–92. DOI: 10.1016/j.vacuum.2016.05.004.
2. Злобин С.К., Михнев М.М., Лаптенко В.Д., Серегин Ю.Н. и др. Автоматизированное оборудование и технология для пайки волноводных трактов космических аппаратов // Вестн. СибГАУ. 2014. № 4. С. 219–229.
3. Шастерик А.А., Данильчик С.С. Электронно-лучевая сварка // Инженерно-педагогическое образование в XXI веке: мат. VIII Республиканской науч.-практич. конф. молодых ученых и студентов. 2018. Ч. 2. С. 225–227.
4. Kaladhar D.S.V.G.K., Pottumuthu B.K., Rao P.V.N., Vadlamudi V., Chaitanya A.K., Reddy R.H. The elements of statistical learning in colon cancer datasets: data mining, inference and prediction. Algorithms Research, 2013, vol. 2, no. 1, pp. 8–17. DOI: 10.5923/j.algorithms.20130201.02.
5. Fernández-Delgado M., Sirsat M.S., Cernadas E., Alawadi S., Barro S., Febrero-Bande M. An extensive experimental survey of regression methods. Neural Networks, 2019, vol. 111, pp. 11–34. DOI: 10.1016/j.neunet.2018.12.010.
6. Albrecht T., Wimmer V., Auinger H.J., Erbe M. et al. Genome-based prediction of testcross values in maize. Theoretical and Applied Genetics, 2011, vol. 123, no. 2, pp. 339–351. DOI: 10.1007/s00122-011-1587-7.
7. Fang T., Lahdelma R. Evaluation of a multiple linear regression model and SARIMA model in forecasting heat demand for district heating system. Applied Energy, 2016, vol. 179, pp. 544–552. DOI: 10.1016/j.apenergy.2016.06.133.
8. Andrews D.F. A Robust method for multiple linear regression. Technometrics, 1974, vol. 16, no. 4, pp. 523–531. DOI: 10.1080/00401706.1974.10489233.
9. Hainmueller J., Hazlett C. Kernel regularized least squares: reducing misspecification bias with a flexible and interpretable machine learning approach. Political Analysis, 2013, no. 22, pp. 143–168. DOI: 10.1093/pan/mpt019.
10. Jayakumar G.S.D.S., Sulthan A. A non-student distribution of jack-knife residuals and identification of outliers. JRSS, 2015, vol. 8, no. 2, pp. 77–96.
11. Chen T., Guestrin C. Xgboost: A scalable tree boosting system. Proc. XXII ACM SIGKDD Intern. Conf., 2016, pp. 785–794. DOI: 10.1145/2939672.2939785.
12. Ke G., Meng Q., Finley T., Wang T. et al. LightGBM: A highly efficient gradient boosting decision tree. NIPS, 2017, vol. 30, pp. 3146–3154.
13. Rizzo M.L. Statistical Computing with R. CRC Press, 2019, 490 p.
14. Denuit M., Hainaut D., Trufin J. Gradient boosting with neural networks. In: Effective Statistical Learning Methods for Actuaries III, Springer Actuarial. Springer, Cham. 2019, pp. 167–192. DOI: 10.1007/978-3-030-25827-6_7.
15. Dierbach C. Python as a first programming language. Journal of Computing Sciences in Colleges, 2014, vol. 29, no. 6, pp. 153–154.
16. Moore A.D. Python GUI Programming with Tkinter: Develop Responsive and Powerful GUI Applications with Tkinter. 2018, 452 p.

17. Hill C. Learning Scientific Programming with Python. Cambridge Univ. Press Publ., 2020, 462 p. DOI: 10.1017/CBO9781139871754.
18. Lutz M. Learning Python: Powerful Object-Oriented Programming. O'Reilly Media Publ., 2013, 1650 p.
19. Guzdial M., Ericson B. Introduction to Computing and Programming in Python. Pearson Publ., 2016, 528 p.
20. Hiam A. Learning BeagleBone Python Programming. Packt Publishing, 2015, 322 p.
21. Jaworski M., Ziadé T. Expert Python Programming: Become a Master in Python by Learning Coding Best Practices and Advanced Programming Concepts in Python 3.7. Packt Publ., 2019, 629 p.

Software & Systems
DOI: 10.15827/0236-235X.134.316-323

Received 22.10.20
2021, vol. 34, no. 2, pp. 316–323

The analysis model and prediction of technological parameters for the process of electron beam welding

V.S. Tynchenko ¹, Ph.D. (Engineering), Associate Professor, vadimond@mail.ru
S.O. Kurashkin ¹, Postgraduate Student, scorpion_ser@mail.ru
I.A. Golovenok ¹, Graduate Student, golovonokia@mail.ru
V.E. Petrenko ¹, Postgraduate Student, dpblra@inbox.ru

¹Academician M.F. Reshetnev Siberian State Aerospace University, Information-Control Systems Chair Krasnoyarsk, 660014, Russian Federation

Abstract. The main purpose of the study is to create a mathematical model for the analysis and prediction of technological parameters of the electron-beam welding process using modern regression models, as well as its implementation as a software system in the Python programming language using Scikit-learn, Pandas, NumPy and Matplotlib software packages.

Actually, the problem of predicting the parameters of the technological process of electron beam welding is a regression problem. There are many algorithms available for solving the regression problem. Under this work, a regression analysis algorithm is used as polynomial regression with $L2$ regularization – ridge regression, as well as an ensemble of decision tree algorithms – a random forest. Using the developed predictive model will allow the technologist to more consciously approach the selection of both the range of variable parameters for research in new technological modes and to improve the quality in already developed technological modes. The application of the proposed methods will also reduce the time and labor costs for the search, development, and adjustment of the technological process.

The paper describes the ridge regression algorithm, as well as an analysis of the applicability of this algorithm to the solution of the problem posed, and also checks the reliability of the forecasts obtained by their direct use. Also, the process of direct training of the model is considered based on data obtained in the experiment framework on the development of the technological process of electron beam welding.

An analysis of the applicability of the approach showed that it is permissible to use the proposed method for technological processes with similar statistical dependences. Implementation of the proposed approach to predicting the parameters of electron beam welding in production will make it possible to support the adoption of technological decisions when working out the technological process of electron beam welding, as well as when put into production new types of products.

Keywords: electron beam welding, technological process, polynomial regression, ensembles of algorithms, machine learning, electron beam, software product, Python, Scikit-learn, Pandas.

Acknowledgements. The reported study was funded by RFBR, Government of the Krasnoyarsk territory and the regional science foundation, project no. 19-48-240007.

References

1. Weglowski M.S., Blacha S., Phillips A. Electron beam welding – Techniques and trends – Review. *Vacuum*, 2016, vol. 130, pp. 72–92. DOI: 10.1016/j.vacuum.2016.05.004.

2. Zlobin S.K., Mikhnev M.M., Laptenok V.D., Seregin Yu.N. et al. Automated equipment and technology for spacecrafts waveguide path soldering. *Vestn. SibSAU*, 2014, no. 4, pp. 219–229 (in Russ.).
3. Shasterik A.A., Danilchik S.S. Electron beam welding. *Proc. VIII Inzhenerno-Pedagogicheskoe Obrazovanie v XXI Veke*, 2018, no. 2, pp. 225–227 (in Russ.).
4. Kaladhar D.S.V.G.K., Pottumuthu B.K., Rao P.V.N., Vadlamudi V., Chaitanya A.K., Reddy R.H. The elements of statistical learning in colon cancer datasets: data mining, inference and prediction. *Algorithms Research*, 2013, vol. 2, no. 1, pp. 8–17. DOI: 10.5923/j.algorithms.20130201.02.
5. Fernández-Delgado M., Sirsat M.S., Cernadas E., Alawadi S., Barro S., Febrero-Bande M. An extensive experimental survey of regression methods. *Neural Networks*, 2019, vol. 111, pp. 11–34. DOI: 10.1016/j.neunet.2018.12.010.
6. Albrecht T., Wimmer V., Auinger H.J., Erbe M. et al. Genome-based prediction of testcross values in maize. *Theoretical and Applied Genetics*, 2011, vol. 123, no. 2, pp. 339–351. DOI: 10.1007/s00122-011-1587-7.
7. Fang T., Lahdelma R. Evaluation of a multiple linear regression model and SARIMA model in forecasting heat demand for district heating system. *Applied Energy*, 2016, vol. 179, pp. 544–552. DOI: 10.1016/j.apenergy.2016.06.133.
8. Andrews D.F. A robust method for multiple linear regression. *Technometrics*, 1974, vol. 16, no. 4, pp. 523–531. DOI: 10.1080/00401706.1974.10489233.
9. Hainmueller J., Hazlett C. Kernel regularized least squares: reducing misspecification bias with a flexible and interpretable machine learning approach. *Political Analysis*, 2013, no. 22, pp. 143–168. DOI: 10.1093/pan/mpt019.
10. Jayakumar G.S.D.S., Sulthan A. A non-student distribution of jack-knife residuals and identification of outliers. *JRSS*, 2015, vol. 8, no. 2, pp. 77–96.
11. Chen T., Guestrin C. Xgboost: A scalable tree boosting system. *Proc. XXII ACM SIGKDD Intern. Conf.*, 2016, pp. 785–794. DOI: 10.1145/2939672.2939785.
12. Ke G., Meng Q., Finley T., Wang T. et al. LightGBM: A highly efficient gradient boosting decision tree. *NIPS*, 2017, vol. 30, pp. 3146–3154.
13. Rizzo M.L. *Statistical Computing with R*. CRC Press, 2019, 490 p.
14. Denuit M., Hainaut D., Trufin J. Gradient boosting with neural networks. In: *Effective Statistical Learning Methods for Actuaries III*, 2019, pp. 167–192. DOI: 10.1007/978-3-030-25827-6_7.
15. Dierbach C. Python as a first programming language. *Journal of Computing Sciences in Colleges*, 2014, vol. 29, no. 6, pp. 153–154.
16. Moore A.D. *Python GUI Programming with Tkinter: Develop Responsive and Powerful GUI Applications with Tkinter*. 2018, 452 p.
17. Hill C. *Learning Scientific Programming with Python*. Cambridge University Press Publ., 2020, 462 p. DOI: 10.1017/CBO9781139871754.
18. Lutz M. *Learning Python: Powerful Object-Oriented Programming*. O'Reilly Media Publ., 2013, 1650 p.
19. Guzdial M., Ericson B. *Introduction to Computing and Programming in Python*. Pearson Publ., 2016, 528 p.
20. Hiam A. *Learning BeagleBone Python Programming*. Packt Publ., 2015, 322 p.
21. Jaworski M., Ziadé T. *Expert Python Programming: Become a Master in Python by Learning Coding Best Practices and Advanced Programming Concepts in Python 3.7*. Packt Publ., 2019, 629 p.

Для цитирования

Тынченко В.С., Курашкин С.О., Головенко И.А., Петренко В.Е. Модель анализа и прогнозирования технологических параметров для процесса электронно-лучевой сварки // Программные продукты и системы. 2021. Т. 34. № 2. С. 316–323. DOI: 10.15827/0236-235X.134.316-323.

For citation

Tynchenko V.S., Kurashkin S.O., Golovenok I.A., Petrenko V.E. The analysis model and prediction of technological parameters for the process of electron beam welding. *Software & Systems*, 2021, vol. 34, no. 2, pp. 316–323 (in Russ.). DOI: 10.15827/0236-235X.134.316-323.