diagnostics

Article

Beyond Henssge's Formula: Using Regression Trees and a Support Vector Machine for Time of Death Estimation in Forensic Medicine

Lívia Mária Dani [1], Dénes Tóth [1,*], Andrew B. Frigyik [2] and Zsolt Kozma [1]

[1] Department of Forensic Medicine, University of Pécs Medical School, Szigeti út 12, H-7624 Pécs, Hungary
[2] Bánki Donát Faculty of Mechanical and Safety Engineering, Óbuda University, Népszínház u. 8, H-1081 Budapest, Hungary; frigyik@gmail.com
* Correspondence: denes.toth@aok.pte.hu

Abstract: Henssge's nomogram is a commonly used method to estimate the time of death. However, uncertainties arising from the graphical solution of the original mathematical formula affect the accuracy of the resulting time interval. Using existing machine learning techniques/tools such as support vector machines (SVMs) and decision trees, we present a more accurate and adaptive method for estimating the time of death compared to Henssge's nomogram. Using the Python programming language, we built a synthetic data-driven model in which the majority of the selected tools can estimate the time of death with low error rates even despite having only 3000 training cases. An SVM with a radial basis function (RBF) kernel and AdaBoost+SVR provided the best results in estimating the time of death with the lowest error with an estimated time of death accuracy of approximately ±20 min or ±9.6 min, respectively, depending on the SVM parameters. The error in the predicted time (t_p[h]) was $t_p \pm 0.7$ h with a 94.45% confidence interval. Because training requires only a small quantity of data, our model can be easily customized to specific populations with varied anthropometric parameters or living in different climatic zones. The errors produced by the proposed method are a magnitude smaller than any previous result.

Keywords: forensic pathology; post mortem interval; multidisciplinary approach; machine learning; support vector machine

Citation: Dani, L.M.; Tóth, D.; Frigyik, A.B.; Kozma, Z. Beyond Henssge's Formula: Using Regression Trees and an SVM for Time of Death Estimation in Forensic Medicine. *Diagnostics* **2023**, *13*, 1260. https://doi.org/10.3390/diagnostics13071260

Academic Editor: Elvira Ventura Spagnolo

Received: 9 March 2023
Revised: 22 March 2023
Accepted: 26 March 2023
Published: 27 March 2023

1. Introduction

The determination of the post mortem interval (PMI) is one of the oldest questions in forensic medicine, which has posed major challenges for experts since its inception and remains the focus of significant research. Both mathematical [1–4] and nonmathematical methods are used to address the problem [5]. The process of changing the temperature of the body involves a complicated interplay of various biological processes and factors, yet it is nevertheless characterized by physical laws. Early studies suggested that the Newtonian cooling law was unsuitable for mathematically characterizing the process because the cooling curve is sigmoidal rather than exponential due to a plateau phase [6–8]. The Marshall–Hoare formula, which was created empirically and contains a linear combination of two exponential functions [9–11], can be used to describe this sigmoidal curve. As it is a transcendental equation, if we want to determine the PMI, we can solve it numerically or graphically with the help of Henssge's nomogram [12–15] or another simplified graphical solution [16]. Subsequent studies required the extension of the Marshall–Hoare formula with a weight-related correction factor [12,17–20], known as the Henssge formula, allowing for a more precise estimation of the PMI.

The mathematical description of the process has not changed since the introduction of the Henssge formula; however, multiple solutions for fitting empirical data using diverse methodologies have been developed: nonlinear least squares [21], conditional probability [22], Bayesian estimation [23,24], finite element simulation [25], Laplace transformation [26], numerical simulations [27–29], and neural networks [30]. A triple exponential model was another strategy to take into account; however, it did not yield the desired outcomes [31,32]. Brute-force calculations [33,34], heat-transfer modeling [25,35,36], the evaluation of a back-calculation [37], and a computational approximation in PHP [38] are other examples of unique approaches.

Machine learning is widely used in numerous fields of medical diagnostics and prognostics [39–42]. One possible method of estimating parameters is finding them by using the method of linear regression. We can choose from various mathematical tools to do so, such as different regression methods, decision trees [43], or applying different neural networks, in which the goal is to find an approximation to the model function belonging to a given learning set. A support vector machine (SVM) can be considered a special neural network, which is a supervised learning method that can have different kernel functions for its decision function [44–46]. The objective of the kernel method is to convert the original problem into a linearly solvable one. With its use, the data describing the problem to be solved are transformed into the kernel space through the application of nonlinear transformations, such as radial basis functions (RBF). The aim of our study was to analyze the accuracy of several different regression methods (decision tree [47], random forests [48], extra trees, bagging, AdaBoost, SVM, AdaBoost + SVM) and their combination in solving the aforementioned mathematical problem using the Python programming language.

The motivation behind this work comes from our desire to support the work of forensic experts by developing a modernized, flexible, and adaptive method that utilizes existing machine learning tools to enable a more accurate estimation of the PMI using present day training data than the commonly used Henssge nomogram and that can adapt to the constantly changing population.

2. Materials and Methods

The Henssge formula, and its graphical solution, the Henssge nomogram, are commonly used in methods to estimate the PMI:

$$\frac{T_r - T_a}{T_0 - T_a} = A \cdot \exp(Bt) + (1 - A) \cdot \exp\left(\frac{AB}{A-1}t\right), \tag{1}$$

where T_r and T_a are the rectal and environmental temperatures, respectively, measured at time t, $T_0 = 37.2$ °C is a constant representing the rectal temperature commonly assumed at the time of death. In the formula, A and B are parameters obtained empirically [17]. The value of the parameter A depends on the environmental temperature (Table 1).

Table 1. The value of the parameter A.

T_a	A
≤23.2 °C	1.25
≥23.3 °C	1.11

The parameter B includes the body weight (m).

$$B = -1.2815 \cdot m^{-0.625} + 0.0284 \tag{2}$$

The Henssge formula in the two temperature ranges is as follows:
For $T_a \leq 23.2$ °C,

$$\frac{T_r - T_a}{37.2 - T_a} = 1.25 \cdot \exp(Bt) - 0.25 \cdot \exp(5Bt); \tag{3}$$

for $T_a \geq 23.3$ °C,

$$\frac{T_r - T_a}{37.2 - T_a} = 1.11 \cdot \exp(Bt) - 0.11 \cdot \exp(10Bt). \tag{4}$$

Any proper mathematical model should be capable of handling the uncertainties that can affect the accuracy of the resulting time of death, the estimate of which is based on the Henssge formula. As can be seen in Henssge's nomogram, uncertainty can be caused by various factors, including the correction factor [49], body weight [12], and a variable environmental temperature and humidity [50]. From a practical point of view, a basic source of error may be the incorrect size ratio of the printed nomogram [51]. The Henssge nomogram graphically handles the uncertainties which can affect the accuracy of the determined PMI, while data-based models incorporate these uncertainties within themselves.

2.1. Data-Driven Model

The purpose of creating the data-driven model was to examine the estimation of the PMI using other mathematical methods. We decided to use decision trees (regression trees) and an SVM with an RBF kernel. Our model relied on the assumption that the generated data from which the system learned closely resembled reality. To create the model and perform the calculations, we used the Python programming language to generate data for learning and testing, which formed the basis of the theoretical model. We chose various regression trees and an SVM from the scikit-learn [52] package.

2.1.1. The Generation of Data and Test Data

For training and testing the regression trees and the SVM, we used generated data. For each parameter that is required for the calculation using the Henssge formula, we randomly selected from a predetermined list of values. These were as follows:

- Time (h): 1–18, with a step of 0.5 h.
- Ambient temperature (°C): -10–35, in increments of 0.5 °C.
- Correction factor: 0.7, 0.9, 1.0, 1.1, 1.2,1.3, 1.4, based on Table 5 in [49]
- Body weight (kg): between 50 and 100 kg, with a precision of 0.5 kg, drawn from a normal distribution with postselection (mean of 70 kg with σ large enough to generate an appropriate quantity of test data close to the upper limit).
- Rectal temperature (°C). Based on the randomly selected data described above, it was calculated from the Henssge formula according to Algorithm 1 which uses Algorithm 2.
- The number of desired data points, which is an approximate value, since some of the weights drawn from a normal distribution were outside of the desired range and therefore were not considered in either the training or test data sets.

According to the literature [49], certain restrictions were taken into account when creating the data:

1. The ambient temperature must not be higher than the measured rectal temperature. Since the rectal temperature was calculated during data generation, this case could not occur.
2. For the correction factors, the value does not need to be adjusted based on weight until 1.4. Beyond this value, it must be corrected, but our model is currently not set up for this (see Table 5 in [49]).
3. In the case of the weight, the selection of the lower and upper limits was again based on the Table 5 in [49]. The selection of the 70 kg average was also based on Table 5 in [49].

Steps for generating the training and test data:

1. Randomly select one parameter set (weight, correction factor, environmental temperature) from the required sets of parameters for the Henssge formula.
2. Determine the rectal temperature by evaluating the Henssge formula.

In the pseudocode of Algorithm 3, the input parameter "count" is an integer that roughly determines the number of generated data points, as values outside the lower and upper bounds of the weight given by the normal distribution were also generated, but they were not used for training and testing.

The samples were separated into training and test data by dividing the total generated data set (X, y) into two parts. Usually, these parts are comprised of different percentages of data, with one part being used as the data to train the model, and the other part being used as the test data to evaluate the performance of the model. X is an $n \times 4$ dimensional matrix, where n is the number of actual data points that meet the conditions, and it contains the selected and calculated parameters in the following order $[m, cf, T_r, T_a]$; y is an n-dimensional vector, where $y_i, i = 1 \ldots n$ contains the randomly selected expected time interval for X_i. During the division, a prespecified percentage of the total data set was chosen for the test data and the rest was used for training.

Algorithm 1 Calculating rectal temperature

```
function CalculateRectalTemperature($t, T_a, k, m$)
    $B_c \leftarrow B(k, m)$
    if $T_a <= 23.2$ then
        return $(37.2 - T_a) \cdot (1.25 \cdot e^{B_c \cdot t} - 0.25 \cdot e^{5 \cdot B_c \cdot t}) + T_a$
    if $T_a >= 23.3$ then
        return $(37.2 - T_a) \cdot (1.11 \cdot e^{c \cdot t} - 0.11 \cdot e^{10 \cdot B_c \cdot t}) + T_a$
```

Algorithm 2 Body weight adjusted by correction factor

```
function B($k, m$)
    return $-1.2815 \cdot (k \cdot m)^{-0.625} + 0.0284$
```

Algorithm 3 Generating training data and test data

```
function GenerateData($count$)
    $times$ ← set of possible elapsed hours
    $ambient\ temperature$ ← set ot possible ambient temperatures in °C
    $correction\ factors$ ← set of possible correction factors
    $body\ weights$ ← set of possible weights in kg, typically with a normal distribution
    while i < count do
        $t$ ← selecting a time randomly from among the $times$
        $T_a$ ← selecting a temperature randomly from among the $ambient\ temperatures$
        $k$ ← selecting a correction factor randomly from among the $correction\ factors$
        $m$ ← selecting a mass randomly from among the $weights$
        /* Calculating rectal temperature */
        $T_r$ ← CalculateRectalTemperature($t, T_a, k, m$)
    storage of generated data
```

2.1.2. Training

We began the process of training our selected regression models by utilizing the partial sample (X_{train}, y_{train}). This data set served as the input for our model training, which was performed using several different approaches.

One of the approaches we utilized was decision (regression) trees, including bagging, random forests, and extremely randomized trees. These techniques have been shown to be effective at modeling complex relationships between variables and making predictions in a variety of scenarios. In addition to the decision trees, we also used support vector regression (SVR) with a radial basis function (RBF) kernel. This is a powerful method that has been shown to be effective at modeling nonlinear relationships between variables. To further refine the results obtained from our extremely randomized trees and SVM, we applied a tree modified with an adaptive boosting method. This allowed us to improve the accuracy and precision of our model predictions.

Once our models had been successfully trained and optimized, we saved them for later use. This ensured that we could quickly and easily access our models and use them to make predictions in new scenarios, without having to repeat the time-consuming and computationally expensive training process.

2.1.3. Testing

We tested the trained model with (X_{test}, y_{test}) subsets. The performance of the model was determined based on the mean absolute error (MAE), mean squared error (MSE), and coefficient of determination (R^2) values (see below). The results of the runs can be found in Appendix A Tables A1–A6.

2.1.4. Error Calculation

There are several mathematical tools available for determining the prediction error. Let N be the size of the sample and $\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_i$ the estimated value of y_i. The residual for the ith observation is defined as $e_i = y_i - \hat{y}_i$, that is, the difference between the expected value y_i and the estimated value for the ith observation.

Sum of Squared Residuals (SSR)

In most cases, we minimized the sum of squared residuals (least-squares method).

$$SSR = e_1^2 + e_2^2 + \cdots + e_N^2 = \sum_{i=1}^{N} (y_i - \hat{y}_i)^2 \tag{5}$$

Mean Squared Error (MSE)

The average of the squares of the differences between the estimated values and the actual values.

$$MSE = \frac{\sum_{i=1}^{N} (y_i - \hat{y}_i)^2}{N} \tag{6}$$

Mean Absolute Error (MAE)

The average of the absolute differences between the estimated and actual values.

$$MAE = \frac{\sum_{i=1}^{N} |y_i - \hat{y}_i|}{N} \tag{7}$$

Coefficient of Determination (R^2)

R^2 is a statistical measure that represents the proportion of the variance in the dependent variable that is predictable from the independent variable(s) in a regression model. R^2 ranges from 0 to 1, where 0 indicates that the model explains none of the variance in the dependent variable, and 1 indicates that the model explains all of the variance.

$$R^2 = \frac{TSS - SSR}{TSS}, \tag{8}$$

where

$$TSS = \sum_{i=1}^{N} \left(y_i - \frac{1}{N} \sum_{i=1}^{N} y_i \right)^2 = \sum_{i=1}^{N} (y_i - \overline{y})^2 \tag{9}$$

is the sum of squared errors, where $\overline{y}$ is the mean value of the given data set.

3. Results

The accuracy of the mathematical model we used for estimating the PMI depended on the proper choice of the relatively large number of adjustable parameters. We considered a choice proper, if we obtained it through a learning process and the resulting model gave meaningful estimates in cases that were similar to those it had already encountered. If we tested a case that fell outside of the domain determined by the learned data, then the estimation error was supposed to grow. The version in this paper used the most commonly used correction factors from the set of all correction factors. The choice of environmental temperature range was based on the Henssge nomograms. One important factor influencing the quality of the generated data was the mean value (70 kg) and standard deviation (σ) of the normal distribution of the body weight, which determined the width of the Gaussian curve, which, in turn, set the range for the random weights obtained. When σ was very small, <5, the generated weights were in a very small range with a very high probability, but if σ was chosen large enough, the data were selected from a larger set. The goal was to have enough data for training with weights between 50 and 100 kg. We determined the σ through multiple trials and for $\approx 11,000$ generated data, and we found that $\sigma = 10$ was already sufficient.

The data were generated such that for each body weight, we were randomly selecting T_a, the correction factor, and the expected time of death using a uniform distribution. Then, T_r was calculated based on these values.

In order to examine the results of the theoretical model, we trained and tested the model using a variety of number of cases and methods, so as to find the tool working with the smallest error for solving the problem.

We designated 25% of the generated data as test data and utilized the remainder for training the system. As the foundation for the theoretical model, we employed various regression tools with differing configurations and sought out the best parameterization for each tool individually. Following this, by using the mathematical tools in combination, we further improved the results obtained. The methods investigated were as follows:

- Regression tree;
- Random forests;
- Extremely randomized trees;
- Tree modified with the bagging method;
- SVR with an RBF kernel;
- SVR improved with adaptive boosting.

Results of Training

The training and estimation time, the errors (*MAE*, *MSE*, R^2), and the best parameterization of the regression tools tested with various parameterizations are presented in the tables in Appendix A Tables A1–A6 for each method. The number of generated data was increased by a thousand, minus the number of cases that did not fall within the determined range of 50–100 kg.

Based on the results, it can be concluded that with a larger training data set, all methods were capable of estimating the time of death with a decreasing error, as shown in the graphs in Figure 1. According to both the MAE, MSE, and the R^2 (≈ 1) values, the best result was achieved by the combined use of SVR and an adaptive regression tree, as this method further improved the results obtained by SVR [53,54]. Since in the used Python implementation, the `C` parameter represents the compromise between minimizing false classification errors and maximizing the decision boundary, meaning the higher the value of `C`, the fewer the false classifications and the stricter the decision margin, we performed four additional control runs with higher `C` values (10, 20, 50, 100) to check the accuracy of the SVR estimate when improved by adaptive boosting for these four cases as well. The results of these were divided into two parts, first for SVR alone, and then for the improved results using the adaptive boosting method.

According to Figure 1, it can be observed that most selected mathematical tools were able to estimate the time of death with low error rates even with a minimum of 3000 training examples, based on the current settings. However, the decision tree was an exception, as it still produced high errors compared to the others, even with over 10,000 data points.

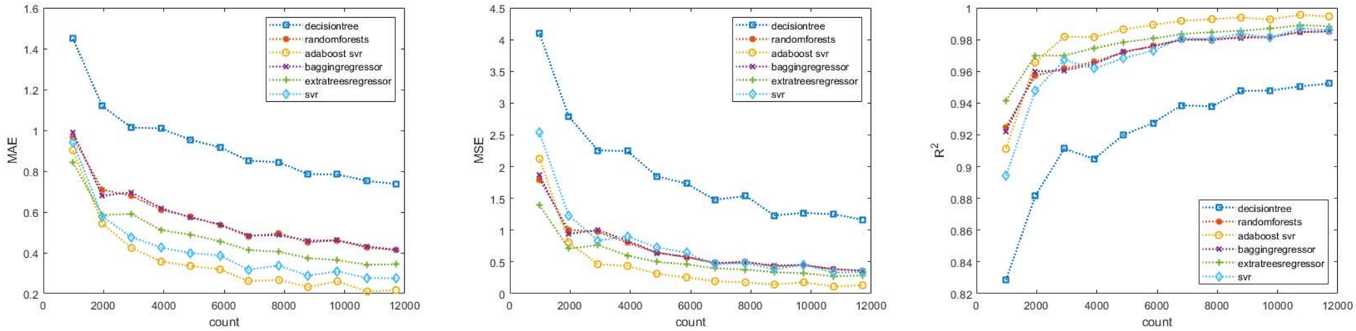

Figure 1. MAE, MSE, and R^2 of theoretical model.

The results obtained with SVR and AdaBoost + SVR models using the parameters `C` = 50 and `C` = 100 at at a sample size of approximately 11,000 were as follows: Based on Tables 2 and 3, it can be concluded that increasing the value of `C` further improved the achieved results. By breaking down the average error of the 25% of test data into correction factors with a 5 kg binning, we determined for the cases of `C` = 5 and `C` = 100 (see in Figure 2) that in the former case, the error was approximately ±0.3 h = 20 min, and in the latter case, the two worst results were approximately ±0.16 h = ±9.6 min, but the average errors were below 4 min.

Table 2. The errors of SVR.

	MAE	MSE	R^2
`C` = 10	0.2578	0.2746	0.9886
`C` = 20	0.2255	0.2252	0.9906
`C` = 50	0.1979	0.1828	0.9924
`C` = 100	0.1683	0.1290	0.9947

Table 3. The errors of AdaBoost + SVR.

	MAE	MSE	R^2
`C` = 10	0.2177	0.1340	0.9944
`C` = 20	0.1875	0.0987	0.9959
`C` = 50	0.1820	0.1109	0.9954
`C` = 100	0.1606	0.0762	0.9969

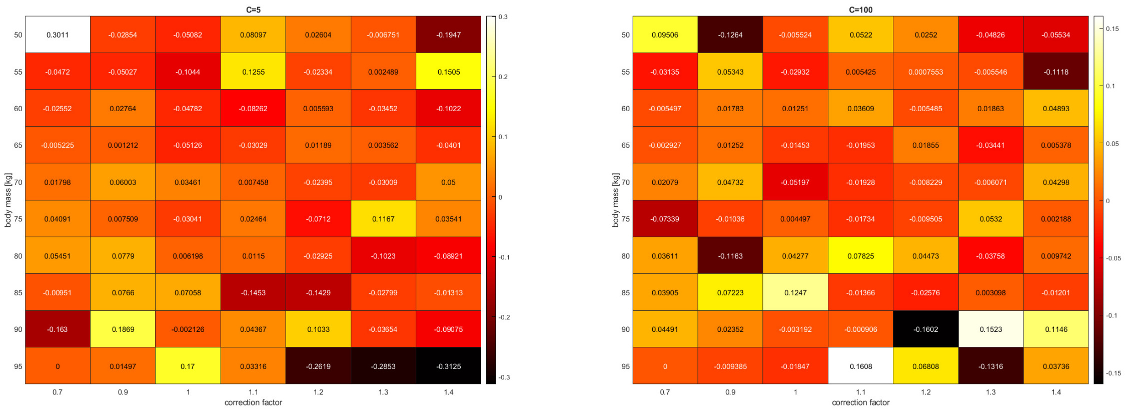

Figure 2. Average error with a 5 kg windowing as a function of the correction factor for `C` = 5 and `C` = 100 cases with the AdaBoost + SVR model.

After comparing the results of various selected methods, it can be concluded that the two best results were obtained with SVR and AdaBoost + SVR, as can be seen in Figure 3 and Table 4. This is due to the fact that these two methods had the most test results within 1σ of the mean.

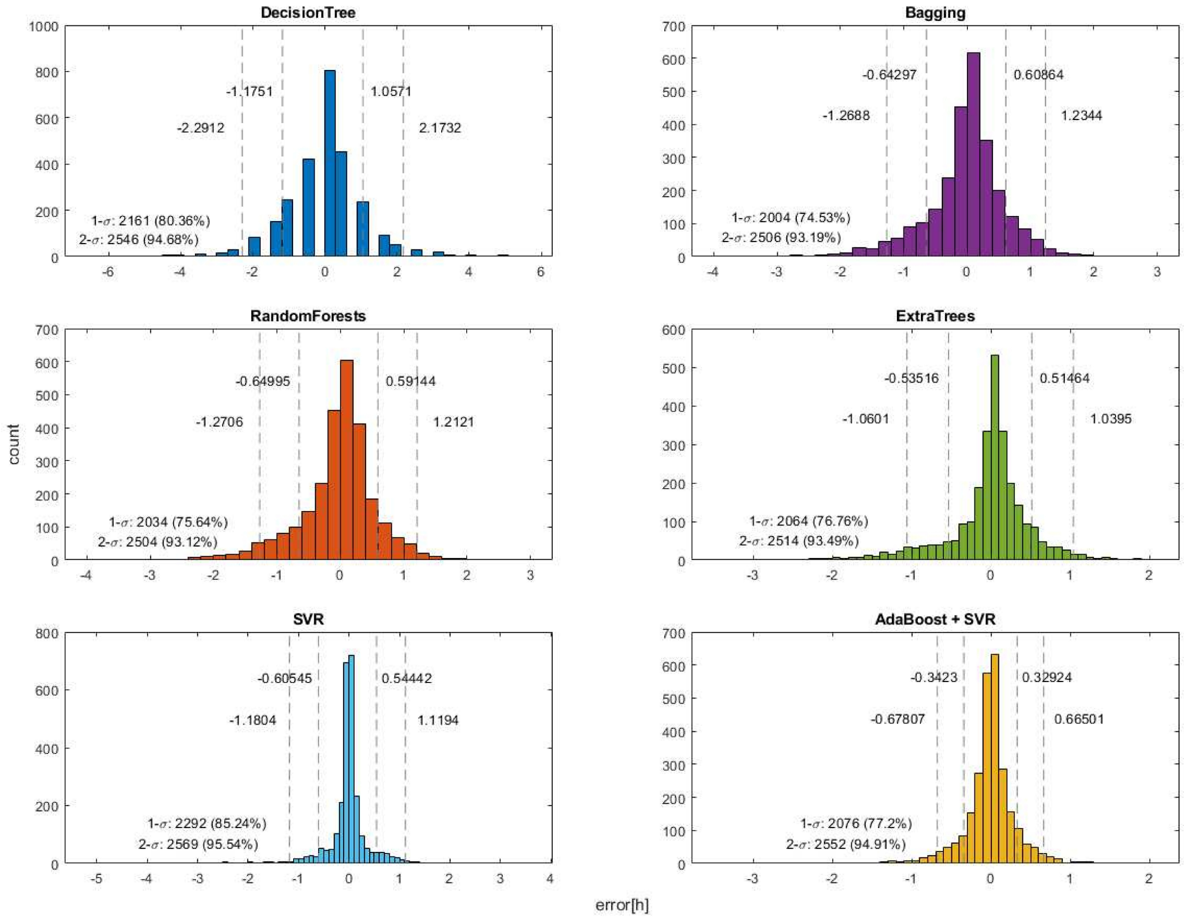

Figure 3. The results of the different methods at distances of 1σ and 2σ.

Table 4. The results of the different methods at distances of 1σ and 2σ.

Name	1σ Value	2σ Value	1σ	2σ
Decision tree	−1.1751–1.0571	−2.2912–2.1732	2161 (80.36%)	2546 (94.68%)
Bagging	−0.64297–0.60864	−1.2688–1.2344	2004 (74.53%)	2506 (93.19%)
Random forests	−0.64995–0.59144	−1.2706–1.2121	2034 (75.64%)	2504 (93.12%)
Extra trees	−0.5316–0.51464	−1.0601–1.0395	2064 (76.76%)	2514 (93.49%)
SVR	−0.60545–0.54442	−1.1804–1.1194	2292 (85.24%)	2569 (95.54%)
AdaBoost + SVR	−0.3423–0.32924	−0.67807–0.66501	2076 (77.2%)	2552 (94.91%)

4. Discussion

The Henssge formula and its graphical solution, the Henssge nomogram, are commonly used to estimate the PMI. However, uncertainties—including the correction factor, body weight, and variable environmental conditions—can affect the accuracy of the resulting time interval. The Henssge nomogram handles these uncertainties graphically, while our model incorporated these uncertainties within itself. In other words, our model did not require the use or knowledge of the Marshall–Hoare or Henssge formula with correction factors, and it did not contain any empirical variables. The generated data closely resembled the reality and formed the basis of our theoretical model. Our data-driven model showed that an SVM with an RBF kernel and the AdaBoost + SVR method provided the best results in estimating the time of death with the lowest error. The estimated accuracy of the time of death was approximately within ±20 min or ±9.6 min, depending on the SVM parameters used. The predicted time error was $t_p \pm 0.7$ h with a 94.45% confidence interval. When compared to the Henssge nomogram, where the accuracy was claimed to be ±2.8 h for both temperature ranges when correction factors were applied, it can be concluded that the created model was capable of estimating the time of death with a sufficient accuracy while taking into account the constraints based on the learned data set. The significant differences and errors arose from the fact that the model encountered these cases with fewer samples during the learning process, but they still fell within the accuracy zone determined by the nomogram. The current limitations of the theoretical model are the number of correction factors, a maximum time interval of 18 h, the need for the training data to be provided with an accuracy of 30 min, and a body weight limited to 50–100 kg. Based on the results presented in Figure 1, it can be inferred that most of the mathematical tools used in this study were able to accurately estimate the time of death with relatively low error rates, even with a minimum of 3000 training examples under the current settings. One notable advantage of our models was that they required very few data for training, which means that they can be applied in various geographical regions, including smaller areas. This feature makes the model highly versatile and adaptable to specific populations with differing anthropometric characteristics or living in different climate zones, because it can be trained with real, available data. Moreover, the model can be easily adapted to suit one's needs, making it an ideal tool for a range of settings and situations.

Most articles on this topic determine the PMI using basic physics or numerical calculations. However, these results cannot be compared to ours because we used the Henssge formula to generate synthetic data. To the best of our knowledge, there is only one paper that used neural networks (multilayer feedforward networks) to address this problem [30]. Zerdazi and coworkers constructed a network using MATLAB 2012 with two layers. The first layer, called the hidden layer, contained 10 neurons, each using the hyperbolic tangent as an activation function. The second layer, known as the output layer, had only one neuron, which employed a linear activation function.

Our method achieved much better results with 257 cases than Henssge's solution. While our model used a different machine learning approach, we can compare our results to theirs (see Tables 5 and 6). To obtain the best comparability, we trained our model again with the same features as those described in that paper [30]. Two scenarios were investigated with common features: an environmental temperature ranging from 4.5 °C to 18 °C, 20% of data for validation, and 20% for testing.

Scenario 1: 275 observations, time of death between 20 min and 18 h. The results obtained are as follows:

Table 5. Comparison of our SVM and AdaBoost + SVM results with the result from the multilayer feedforward network (neural method) by Zerdazi et al. [30] in case of Scenario 1.

Name	MAE	MSE
Neural method	1.85	5.69
SVR	0.17	0.14
AdaBoost + SVR	0.17	0.12

Scenario 2: 184 observations, time of death less than 7 h. The results obtained are as follows:

Table 6. Comparison of our SVM and AdaBoost + SVM results with the result of the multilayer feedforward network (neural method) by Zerdazi et al. [30] in case of Scenario 2.

Name	MAE	MSE
Neural method	0.86	1.21
SVR	0.18	0.08
AdaBoost + SVR	0.14	0.05

From these results, we can conclude that the proposed method performed much better: all the errors were at least one order of magnitude smaller than the results in [30].

Further testing with real data is needed. The future goal is to develop a phone or web application based on the model with a graphical interface for easier use and to create a database for storing anonymized training data.

5. Conclusions

Our research demonstrated that the estimated PMIs produced by our models using existing machine learning tools such as SVMs and decision trees were far more satisfactory than those produced by the Henssge formula or the method utilizing neural networks. In contrast to traditional mathematical methods, including the Henssge nomogram, that yield fixed formulas and whose performance remains constant, our models can be continuously improved because training can be resumed whenever new additional data are available. As our models estimated the PMI with low error rates even with only 3000 training cases, they can be easily adapted to specific populations with different characteristics or living in different climatic zones.

Author Contributions: Conceptualization, L.M.D., D.T., and A.B.F.; methodology, A.B.F.; software, L.M.D.; validation, L.M.D., D.T., and A.B.F.; writing—original draft, L.M.D.; writing—review and editing, L.M.D., D.T., A.B.F., and Z.K.; visualization, L.M.D. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The source code of our model is available: https://github.com/livdan/TOD (accessed on 6 March 2023).

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

MAE	Mean absolute error
MSE	Mean squared error
PMI	Post mortem interval
R^2	Coefficient of determination
RBF	Radial basis function
SSR	Sum of squared residuals
SVM	Support vector machine
SVR	Support vector regression
TSS	Sum of squared errors

Appendix A

Table A1. Results of running the model with: DecisionTreeRegressor.

(#)	Case Number	Training Time (s)	Prediction Time (s)	MAE	MSE	R2	Best Parameters
1	986	3.210	0.028	1.4494	4.1032	0.8285	random_state = 10
2	1955	4.079	0.035	1.1186	2.7904	0.8814	criterion = ‘friedman_mse’
3	2925	4.773	0.085	1.0143	2.2565	0.9114	criterion = ‘friedman_mse’, random_state = 10
4	3909	3.410	0.066	1.0092	2.2439	0.9048	criterion = ‘poisson’, random_state = 10
5	4881	3.994	0.084	0.9533	1.8468	0.9199	criterion = ‘friedman_mse’, random_state = 50
6	5864	3.757	0.094	0.9168	1.7343	0.9275	criterion = ‘friedman_mse’, random_state = 10
7	6806	5.299	0.106	0.8515	1.4768	0.9386	default
8	7824	5.210	0.122	0.8436	1.5394	0.9380	criterion = ‘friedman_mse’, random_state = 100
9	8785	4.753	0.139	0.7854	1.2311	0.9476	random_state = 50
10	9759	5.269	0.150	0.7842	1.2724	0.9480	criterion = ‘friedman_mse’, random_state = 50
11	10755	4.995	0.166	0.7522	1.2489	0.9505	criterion = ‘poisson’, random_state = 25
12	11708	6.811	0.184	0.7371	1.1654	0.9523	criterion = ‘poisson’, random_state = 10

Table A2. Results of running the model with: RandomForestRegressor.

(#)	Case Number	Training Time (s)	Prediction Time (s)	MAE	MSE	R2	Best Parameters
1	986	39.188	2.749	0.9662	1.7966	0.9249	criterion = ‘poisson’, max_features = None, n_estimators = 200, random_state = 10
2	1955	61.648	2.630	0.7092	1.0064	0.9572	max_features = None, random_state = 25
3	2925	106.174	8.661	0.6797	0.9713	0.9619	criterion = ‘poisson’, max_features = None, n_estimators = 200, random_state = 25
4	3909	120.701	9.031	0.6110	0.8032	0.9659	criterion = ‘poisson’, max_features = None, n_estimators = 150, random_state = 25
5	4881	148.617	14.466	0.5761	0.6445	0.9720	criterion = ‘poisson’, max_features = None, n_estimators = 200, random_state = 25
6	5864	154.820	15.039	0.5370	0.5759	0.9759	max_features = None, n_estimators = 200, random_state = 50
7	6806	180.887	15.659	0.4801	0.4774	0.9802	criterion = ’friedman_mse’, max_features = None, n_estimators = 150
8	7824	207.772	15.076	0.4967	0.5081	0.9795	criterion = ‘poisson’, max_features = None, n_estimators = 150
9	8785	266.933	24.548	0.4510	0.4271	0.9818	criterion = ‘poisson’, max_features = None, n_estimators = 200
10	9759	342.604	31.534	0.4608	0.4486	0.9817	criterion = ‘poisson’, max_features = None, n_estimators = 200, random_state = 10
11	10755	401.349	31.706	0.4252	0.3816	0.9849	criterion = ’friedman_mse’, max_features = None, n_estimators = 200, random_state = 10
12	11708	344.101	29.897	0.4134	0.3625	0.9852	criterion = ‘poisson’, max_features = None, n_estimators = 200, random_state = 10

Table A3. Results of running the model with: ExtraTreesRegressor.

(#)	Case Number	Training Time (s)	Prediction Time (s)	MAE	MSE	R2	Best Parameters
1	986	12.355	0.883	0.8422	1.3935	0.9417	max_features = None, n_estimators = 50
2	1955	24.224	5.954	0.5849	0.7086	0.9699	criterion = ‘friedman_mse’, max_features = None, n_estimators = 200
3	2925	29.064	9.108	0.5909	0.7653	0.9700	criterion = ‘friedman_mse’, max_features = None, n_estimators = 200
4	3909	40.331	13.009	0.5108	0.5983	0.9746	criterion = ‘friedman_mse’, max_features = None, n_estimators = 200
5	4881	48.345	10.611	0.4896	0.5011	0.9783	criterion = ‘poisson’, max_features = None, n_estimators = 125
6	5864	60.539	17.568	0.4568	0.4608	0.9807	criterion = ‘poisson’, max_features = None, n_estimators = 200
7	6806	67.474	11.464	0.4143	0.3983	0.9834	criterion = ‘poisson’, max_features = None, n_estimators = 125
8	7824	81.665	23.157	0.4052	0.3798	0.9847	max_features = None, n_estimators = 200
9	8785	103.041	15.619	0.3745	0.3372	0.9856	criterion = ‘poisson’, max_features = None, n_estimators = 125
10	9759	132.059	32.412	0.3645	0.3218	0.9869	criterion = ‘friedman_mse’, max_features = None, n_estimators = 200
11	10755	144.701	33.263	0.3419	0.2745	0.9891	max_features = None, n_estimators = 200
12	11708	139.604	30.451	0.3437	0.2850	0.9883	max_features = None, n_estimators = 200

Table A4. Results of running the model with: BaggingRegressor.

(#)	Case Number	Training Time (s)	Prediction Time (s)	MAE	MSE	R2	Best Parameters
1	986	2.176	1.593	0.9907	1.8638	0.9221	n_estimators = 100
2	1955	3.515	4.360	0.6796	0.9420	0.9600	n_estimators = 125
3	2925	4.198	4.913	0.6961	1.0027	0.9606	n_estimators = 100
4	3909	6.989	13.231	0.6193	0.8344	0.9646	n_estimators = 200
5	4881	7.111	10.565	0.5726	0.6419	0.9722	n_estimators = 125
6	5864	7.989	18.109	0.5387	0.5820	0.9757	n_estimators = 200
7	6806	6.822	10.743	0.4857	0.4862	0.9798	n_estimators = 100
8	7824	9.390	24.057	0.4859	0.4912	0.9802	n_estimators = 200
9	8785	9.695	19.855	0.4592	0.4441	0.9811	n_estimators = 150
10	9759	9.830	17.622	0.4616	0.4524	0.9815	n_estimators = 150
11	10755	11.064	24.376	0.4313	0.3870	0.9847	n_estimators = 150
12	11708	11.208	24.229	0.4153	0.3629	0.9851	n_estimators = 200

Table A5. Results of running the model with: SVR.

(#)	Case Number	Training Time (s)	Prediction Time (s)	MAE	MSE	R2	Best Parameters
1	986	0.749	0.070	0.9406	2.5332	0.8941	C = 5, epsilon = 0.005, gamma = 1
2	1955	3.184	0.163	0.5779	1.2309	0.9477	C = 5, epsilon = 0.01, gamma = 1
3	2925	5.612	0.333	0.4774	0.8372	0.9671	C = 5, epsilon = 0.01, gamma = 1
4	3909	9.611	0.360	0.4257	0.9012	0.9618	C = 5, epsilon = 0.05, gamma = 1
5	4881	13.404	0.626	0.3987	0.7310	0.9683	C = 5, epsilon = 0.05, gamma = 1
6	5864	18.982	0.612	0.3873	0.6430	0.9731	C = 5, epsilon = 0.05, gamma = 1
7	6806	28.362	1.058	0.3181	0.4621	0.9808	C = 5, epsilon = 0.01, gamma = 1
8	7824	28.212	0.824	0.3372	0.4831	0.9805	C = 5, gamma = 1
9	8785	36.948	1.110	0.2882	0.3819	0.9837	C = 5, epsilon = 0.05, gamma = 1
10	9759	50.228	1.318	0.3093	0.4550	0.9814	C = 5, epsilon = 0.05, gamma = 1
11	10755	58.958	1.651	0.2763	0.3306	0.9869	C = 5, epsilon = 0.01, gamma = 1
12	11708	70.528	2.318	0.2753	0.3388	0.9861	C = 5, epsilon = 0.01, gamma = 2

Table A6. Results of running the model with:AdaBoostRegressor + SVR.

(#)	Case Number	Training Time (s)	Prediction Time (s)	MAE	MSE	R2	Best Parameters
1	986	30.903	1.401	0.9026	2.1245	0.9112	loss = ‘exponential’, n_estimators = 20, random_state = 15
2	1955	132.368	2.591	0.5437	0.8068	0.9657	loss = ‘exponential’, n_estimators = 20, random_state = 15
3	2925	262.288	3.775	0.4243	0.4630	0.9818	loss = ’square’, n_estimators = 20
4	3909	474.196	7.863	0.3575	0.4367	0.9815	loss = ‘exponential’, n_estimators = 20
5	4881	737.671	9.702	0.3358	0.3134	0.9864	loss = ‘exponential’, n_estimators = 20, random_state = 15
6	5864	1163.730	13.094	0.3190	0.2559	0.9893	loss = ‘exponential’, n_estimators = 20, random_state = 25
7	6806	1507.816	16.351	0.2619	0.1962	0.9918	loss = ‘exponential’, n_estimators = 20, random_state = 10
8	7824	1778.717	19.738	0.2667	0.1764	0.9929	loss = ‘exponential’, n_estimators = 20
9	8785	2378.614	25.049	0.2325	0.1434	0.9939	loss = ‘exponential’, n_estimators = 20, random_state = 25
10	9759	3199.345	31.656	0.2607	0.1788	0.9927	n_estimators = 20, random_state = 25
11	10755	3350.830	33.012	0.2102	0.1115	0.9956	loss = ‘exponential’, n_estimators = 20, random_state = 25
12	11708	4460.481	44.018	0.2177	0.1323	0.9946	loss = ’square’, n_estimators = 20, random_state = 2

References

1. Knight, B. The evolution of methods for estimating the time of death from body temperature. *Forensic Sci. Int.* **1988**, *36*, 47–55. [CrossRef] [PubMed]
2. Nokes, L.; Flint, T.; Williams, J.; Knight, B. The application of eight reported temperature-based algorithms to calculate the postmortem interval. *Forensic Sci. Int.* **1992**, *54*, 109–125. [CrossRef] [PubMed]
3. Madea, B. Methods for determining time of death. *Forensic Sci. Med. Pathol.* **2016**, *12*, 451–485. [CrossRef] [PubMed]
4. Laplace, K.; Baccino, E.; Peyron, P.A. Estimation of the time since death based on body cooling: A comparative study of four temperature-based methods. *Int. J. Leg. Med.* **2021**, *135*, 2479–2487. [CrossRef]
5. Mathur, A.; Agrawal, Y. An overview of methods used for estimation of time since death. *Aust. J. Forensic Sci.* **2011**, *43*, 275–285. [CrossRef]
6. Rainy, H. On the cooling of dead bodies as indicating the length of time since death. *Glasg. Med. J.* **1868**, *1*, 323–330.
7. Brown, A.; Marshall, T. Body temperature as a means of estimating the time of death. *Forensic Sci.* **1974**, *4*, 125–133. [CrossRef]
8. Al-Alousi, L.M. A study of the shape of the post-mortem cooling curve in 117 forensic cases. *Forensic Sci. Int.* **2002**, *125*, 237–244. [CrossRef]
9. Marshall, T. K. Estimating the time since death - the rectal cooling after death and its mathematical representation. *J. Forensic Sci.* **1962**, *7*, 56–81.
10. Marshall, T.K. The use of the cooling formula in the study of post mortem body cooling. *J. Forensic Sci.* **1962**, *7*, 189– 210.
11. Marshall, T.K. The use of body temperature in estimating the time of death. *J. Forensic Sci.* **1962**, *7*, 211–221.
12. Henssge, C. Death time estimation in case work. I. The rectal temperature time of death nomogram. *Forensic Sci. Int.* **1988**, *38*, 209–236. [CrossRef]
13. Henssge, C.; Madea, B.; Gallenkemper, E. Death time estimation in case work. II. Integration of different methods. *Forensic Sci. Int.* **1988**, *39*, 77–87. [CrossRef] [PubMed]
14. Henßge, C.; Madea, B. Estimation of the time since death in the early post-mortem period. *Forensic Sci. Int.* **2004**, *144*, 167–175. [CrossRef] [PubMed]
15. Leinbach, C. Beyond Newton's law of cooling - estimation of time since death. *Int. J. Math. Educ. Sci. Technol.* **2011**, *42*, 765–774. [CrossRef]
16. Potente, S.; Kettner, M.; Verhoff, M.; Ishikawa, T. Minimum time since death when the body has either reached or closely approximated equilibrium with ambient temperature. *Forensic Sci. Int.* **2017**, *281*, 63–66. [CrossRef]
17. Henßge, C. Die Präzision von Todeszeitschätzungen durch die mathematische Beschreibung der rektalen Leichenabkühlung. *Z. FÜR Rechtsmed.* **1979**, *83*, 49–67. [CrossRef]
18. Henssge, C. Todeszeitschätzungen durch die mathematische Beschreibung der rektalen Leichenabkühlung unter verschiedenen Abkühlungsbedingungen. *Z. FüR Rechtsmed.* **1981**, *87*, 147–178. [CrossRef]
19. Hubig, M.; Muggenthaler, H.; Sinicina, I.; Mall, G. Body mass and corrective factor: Impact on temperature-based death time estimation. *Int. J. Legal. Med.* **2011**, *125*, 437–444. [CrossRef]
20. Potente, S.; Kettner, M.; Ishikawa, T. Time since death nomographs implementing the nomogram, body weight adjusted correction factors, metric and imperial measurements. *Int. J. Leg. Med.* **2019**, *133*, 491–499. [CrossRef]
21. Rodrigo, M.R. A Nonlinear Least Squares Approach to Time of Death Estimation Via Body Cooling. *J. Forensic Sci.* **2016**, *61*. [CrossRef] [PubMed]
22. Biermann, F.M.; Potente, S. The deployment of conditional probability distributions for death time estimation. *Forensic Sci. Int.* **2011**, *210*, 82–86. [CrossRef] [PubMed]
23. Hubig, M.; Muggenthaler, H.; Mall, G. Conditional probability distribution (CPD) method in temperature based death time estimation: Error propagation analysis. *Forensic Sci. Int.* **2014**, *238*, 53–58. [CrossRef]
24. Giana, F.E.; Onetto, M.A.; Pregliasco, R.G. Uncertainty in the estimation of the postmortem interval based on rectal temperature measurements: A Bayesian approach. *Forensic Sci. Int.* **2020**, *317*, 110505. [CrossRef]
25. Bartgis, C.; LeBrun, A.; Ma, R.; Zhu, L. Determination of Time of Death in Forensic Science via a 3-D Whole Body Heat Transfer Model. *J. Therm. Biol.* **2016**, *62*, 109–115. [CrossRef] [PubMed]
26. Rodrigo, M.R. Time of death estimation from temperature readings only: A Laplace transform approach. *Appl. Math. Lett.* **2015**, *39*, 47–52. [CrossRef]
27. Muñoz-Barús, J.I.; Rodríguez-Calvo, M.S.; Suárez-Peñaranda, J.M.; Vieira, D.N.; Cadarso-Suárez, C.; Febrero-Bande, M. PMICALC: An R code-based software for estimating post-mortem interval (PMI) compatible with Windows, Mac and Linux operating systems. *Forensic Sci. Int.* **2010**, *194*, 49–52. [CrossRef] [PubMed]
28. Nedugov, G.V. Numerical method for solving double exponential models of corpse cooling in the determination of the time of death. *Sud. Med. Ekspert* **2021**, *64*, 25–28. [CrossRef]
29. Abraham, J.; Wei, T.; Cheng, L. Validation of a new method of providing case-specific time-of-death estimates using cadaver temperatures. *J. Forensic Sci.* **2023**, *Early View*. [CrossRef]
30. Zerdazi, D.; Chibat, A.; Rahmani, F.L. Estimation of Postmortem Period by Means of Artificial Neural Networks. *Electron. J. Appl. Stat. Anal.* **2016**, *9*, 326.

31. Al-Alousi, L.M.; Anderson, R.A.; Worster, D.M.; Land, D.V. Factors influencing the precision of estimating the postmortem interval using the triple-exponential formulae (TEF): Part I. A study of the effect of body variables and covering of the torso on the postmortem brain, liver and rectal cooling rates in 117 forensic cases. *Forensic Sci. Int.* **2002**, *125*, 223–230. [CrossRef]
32. Al Alousi, L.M.; Anderson, R.A.; Worster, D.M.; Land, D.V. Factors influencing the precision of estimating the postmortem interval using the triple-exponential formulae (TEF): Part II. A study of the effect of body temperature at the moment of death on the postmortem brain, liver and rectal cooling in 117 forensic cases. *Forensic Sci. Int.* **2002**, *125*, 231–236. [CrossRef]
33. Potente, S.; Henneicke, L.; Schmidt, P. Prism—A novel approach to dead body cooling and its parameters. *Forensic Sci. Int.* **2021**, *325*, 110870. [CrossRef]
34. Potente, S.; Henneicke, L.; Schmidt, P. Prism (II): 127 cooling dummy experiments. *Forensic. Sci. Int.* **2022**, *333*, 111238. [CrossRef] [PubMed]
35. Wilk, L.S.; Hoveling, R.J.M.; Edelman, G.J.; Hardy, H.J.J.; van Schouwen, S.; van Venrooij, H.; Aalders, M.C.G. Reconstructing the time since death using noninvasive thermometry and numerical analysis. *Sci. Adv.* **2020**, *6*, eaba4243. [CrossRef]
36. Sharma, P.; Kabir, C.S. A Simplified Approach to Understanding Body Cooling Behavior and Estimating the Postmortem Interval. *Forensic Sci.* **2022**, *2*, 403–416. [CrossRef]
37. Bovenschen, M.; Schwender, H.; Ritz-Timme, S.; Beseoglu, K.; Hartung, B. Estimation of time since death after a post-mortem change in ambient temperature: Evaluation of a back-calculation approach. *Forensic. Sci. Int.* **2020**, *319*, 110656. [CrossRef] [PubMed]
38. Schweitzer, W.; Thali, M.J. Computationally approximated solution for the equation for Henssge's time of death estimation. *Bmc Med. Inform. Decis. Mak.* **2019**, *19*, 201. [CrossRef]
39. Franchuk, V.; Mikhaylichenko, B.; Franchuk, M. Application of the decision tree method in forensic-medical practice in the analysis of 'doctors proceedings. *Sud.-Meditsinskaia Ekspertiza* **2020**, *63*, 9–14. [CrossRef] [PubMed]
40. Lenin Fred, A.; Kumar, S.; Padmanabhan, P.; Gulyas, B.; Haridhas, A.K.; Dayana, N. Chapter 8— Multiview decision tree-based segmentation of tumors in MR brain medical images. In *Handbook of Decision Support Systems for Neurological Disorders*; Jude, H.D., Ed.; Academic Press: Cambridge, MA, USA, 2021; pp. 125–142. [CrossRef]
41. Murdaca, G.; Caprioli, S.; Tonacci, A.; Billeci, L.; Greco, M.; Negrini, S.; Cittadini, G.; Zentilin, P.; Ventura Spagnolo, E.; Gangemi, S. A Machine Learning Application to Predict Early Lung Involvement in Scleroderma: A Feasibility Evaluation. *Diagnostics* **2021**, *11*, 1880. [CrossRef]
42. Shehab, M.; Abualigah, L.; Shambour, Q.; Abu-Hashem, M.A.; Shambour, M.K.Y.; Alsalibi, A.I.; Gandomi, A.H. Machine learning in medical applications: A review of state-of-the-art methods. *Comput. Biol. Med.* **2022**, *145*, 105458. [CrossRef] [PubMed]
43. Gareth, J.; Witten, D.; Hastie, T.; Tibshirani, R. *An Introduction to Statistical Learning, with Applications in R*; Springer: New York, NY, USA, 2021; p. 607. [CrossRef]
44. Vapnik, V.N. *The Nature of Statistical Learning Theory*, 2nd ed.; Springer: New York, NY, USA, 2000.
45. Haykin, S. *Neural Networks and Learning Machines*; Number 10. k. in Neural Networks and Learning Machines; Prentice Hall: Upper Saddle River, NJ, USA, 2009.
46. Chen, B.B. *Comprehensive Chemometrics, Chemical and Biochemical Data Analysis; Elsevier: Amsterdam, The Netherlands, 2009; Volume 3.*
47. Geurts, P.; Ernst, D.; Wehenkel, L. Extremely Randomized Trees. *Mach. Learn.* **2006**, *36*, 3–42. [CrossRef]
48. Breiman, L. Random Forests. *Mach. Learn.* **2001**, *45*, 5–32. [CrossRef]
49. Henssge, C. Rectal temperature time of death nomogram: Dependence of corrective factors on the body weight under stronger thermic insulation conditions. *Forensic. Sci. Int.* **1992**, *54*, 51–66. [CrossRef]
50. Mall, G.; Hubig, M.; Eckl, M.; Buettner, A.; Eisenmenger, W. Modelling postmortem surface cooling in continuously changing environmental temperature. *Leg. Med.* **2002**, *4*, 164–173. [CrossRef]
51. Burger, E.H.; Dempers, J.J.; Steiner, S.; Shepherd, R. Henssge nomogram typesetting error. *Forensic. Sci. Med. Pathol.* **2013**, *9*, 615–617. [CrossRef] [PubMed]
52. scikit-learn 1.2.1. Available online: https://scikit-learn.org/stable/10.02.2023 (accessed on 14 September 2022).
53. Hastie, T.J.; Rosset, S.; Zhu, J.; Zou, H. Multi-class AdaBoost. *Stat. Its Interface* **2009**, *2*, 349–360. [CrossRef]
54. Freund, Y.; Schapire, R.E. A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. *J. Comput. Syst. Sci.* **1997**, *55*, 119–139. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.