J. Dairy Sci. 105
https://doi.org/10.3168/jds.2021-21663

Machine learning classification of breeding protocol descriptions from Canadian Holsteins

L. M. Alcantara,[1] F. S. Schenkel,[1]* C. Lynch,[1] G. A. Oliveira Junior,[1] C. F. Baes,[1,2] and D. Tulpan[1]
[1]Centre for Genetic Improvement of Livestock, Department of Animal Biosciences, University of Guelph, Guelph, Ontario N1G 2W1, Canada
[2]Institute of Genetics, Department of Clinical Research and Veterinary Public Health, University of Bern, Bern, 3001, Switzerland

ABSTRACT

Dairy farmers are motivated to ensure cows become pregnant in an optimal and timely manner. Although timed artificial insemination (TAI) is a successful management tool in dairy cattle, it masks an animal's innate fertility performance, likely reducing the accuracy of genetic evaluations for fertility traits. Therefore, separating fertility traits based on the recorded management technique involved in the breeding process or adding the breeding protocol as an effect to the model can be viable approaches to address the potential bias caused by such management decisions. Nevertheless, there is a lack of specificity and uniformity in the recording of breeding protocol descriptions by dairy farmers. Therefore, this study investigated the use of 8 supervised machine learning algorithms to classify 1,835 unique breeding protocol descriptions from 981 herds into the following 2 classes: TAI or other than TAI. Our results showed that models that used a stacking classifier algorithm had the highest Matthews correlation coefficient (0.94 ± 0.04, mean ± SD) and maximized precision and recall (F1-score = 0.96 ± 0.03) on test data. Nonetheless, their F1-scores on test data were not different from 5 out of the other 7 algorithms considered. Altogether, results presented herein suggest machine learning algorithms can be used to produce robust models that correctly identify TAI protocols from dairy cattle breeding records, thus opening the opportunity for unbiased genetic evaluation of animals based on their natural fertility.

Key words: breeding protocol description, Canadian Holstein, machine learning classifier, timed artificial insemination

INTRODUCTION

Dairy farmers are motivated to ensure cows become pregnant in an optimal and timely manner (Ribeiro et al., 2012). Artificial insemination programs are extensively used in the dairy industry, which makes estrus detection crucial for successful breeding (Roelofs et al., 2010; Silper et al., 2017). In addition to the difficulty of the task itself, estrus detection is greatly affected by management-related factors such as increased herd size, animal density, time standing on concrete, and limited number of qualified staff on dairy farms (Vailes and Britt, 1990; Denis-Robichaud et al., 2016). Hormonal synchronization protocols, also known as timed AI (**TAI**) protocols, are commonly used on dairy farms to increase overall herd conception rates (Ribeiro et al., 2012). Such protocols alleviate the pressure of estrus detection by making ovulation time easier to predict, as TAI relies on hormones to synchronize follicle growth, corpus luteum regression, and ovulation (Cerri et al., 2004).

A recent study (Lynch et al., 2021) with Canadian Holsteins looked at breeding data extracted from official records and found that 60% of studied herds used TAI in 2017, whereas approximately 20% of all animals were on TAI in this same year. They also reported that around 10% of herds use TAI protocols on more than 50% of their animals. Although TAI has been a successful management tool in dairy cattle, it was reported that it potentially introduces bias in genetic breeding programs, as previously demonstrated in a simulation study (Oliveira Junior et al., 2021) and with on-farm data (Lynch et al., 2021). Both studies showed considerable reranking of bulls when EBV were calculated with and without TAI records, showing evidence that TAI masks an animal's true fertility performance, thus likely adding bias to genetic evaluations of fertility traits. One of the suggestions proposed by Lynch et al. (2021) to address this bias was to account for the effect of TAI in the genetic evaluation model for fertility or to split current fertility traits by the management technique used for the breeding.

Received December 3, 2021.
Accepted June 8, 2022.

*Corresponding author: schenkel@uoguelph.ca

Official genetic evaluation for fertility traits in Canada does not use breeding records extracted directly from herd-management software. Instead, breeding records are provided by AI companies and accredited farms to Lactanet (Canadian Network of Dairy Excellence) through a Data Exchange System (**DES**; Lactanet, 2022a). The type of service in the DES insemination record layout includes inseminations performed by AI technicians or herd owners (accredited farms only), natural supervised breeding, pasture natural unsupervised breeding, and embryo transfer if the cow was a recipient (Lactanet, 2022b). Consequently, data used by Lactanet would match to either TAI or heat detection protocols only. Nonetheless, insemination records provided through the DES do not include the breeding protocol description used. Therefore, this information should be retrieved from herd management software and added to the national database.

However, Lynch et al. (2021) highlighted that there is no standardization for Canadian farmers to record breeding protocol information. To keep track of the reproductive performance of their herd, farmers record the breeding protocol description (**BPD**) used for each breeding, typically consisting of a combination of only a few keywords, abbreviations, numbers, or special characters, such as "Natural," "Ovsynch," "Standing," "GGPG," and "CIDR." However, because there is no standard for creating these BPD, 2 farms might use different BPD for the same or variations of the same protocol. For instance, Lynch et al. (2021) reported that there were 156 unique BPD describing "Ovsynch" in their dataset, from which the source of variation came from typos, capitalization, or abbreviations. Furthermore, they found almost 6,000 different BPD across 1,192 herds, of which 2,021 were herd specific. This highlights the lack of specificity and uniformity in the recording of BPD by farmers, making it extremely difficult to properly classify the description of the breeding method using deterministic or traditional stochastic methods.

Nevertheless, methods based on text mining (e.g., text classification) are promising for the classification of BPD. One traditional approach for text classification is to use the bag-of-words representation, which associates a text with a vector indicating the number of occurrences of words from a predefined dictionary (HaCohen-Kerner et al., 2020). Breeding protocol descriptions are normally short descriptions, rather than a full sentence; therefore, using character bigrams rather than *n* combinations of words seem to be a more appropriate approach for the classification of BPD. As reviewed by Lecluze et al. (2013), the idea of considering character N-grams rather than words has been successfully applied on many tasks, such as author identification, language identification, speech analysis, text categorization, numerical classification of multilingual documents, information retrieval, and multilingual automatic alignment. Thus, throughout this manuscript, the terms feature and character bigram will be used interchangeably.

To our knowledge, no previous work has been done to automate, with or without machine learning algorithms, the identification of BPD that corresponds to TAI protocols. However, there are publications available describing similar problems of text classification focused on record matching and its implication within the agriculture sector. As an example, Aiken et al. (2019) used deterministic, stochastic, and machine learning methods to apply and compare data linkage in the absence of a unique universal farm identifier. Among the methods used, they reported supervised and unsupervised algorithms (support vector machine and bagged clustering, respectively) to accurately match strings (99.9%), where the Levenshtein distance (Haldar and Mukhopadhyay, 2011) was the best metric.

We hypothesized that machine learning would be a viable approach to build robust models able to identify BPD corresponding to hormonal synchronization protocols. Therefore, a supervised learning approach was implemented in this study to classify BPD used with Canadian Holsteins into the following 2 categories: TAI protocols or not TAI protocols (**NTAI**).

MATERIALS AND METHODS

This study used historical data generated from the day-to-day operations of farms across Canada enrolled in Dairy Herd Improvement services. All dairy farms in Canada must adhere to the code of practice for the care and handling of dairy cattle by the National Farm Animal Care Council of Canada (Lacombe, Alberta).

Breeding Code Data

Breeding data recorded by Canadian Holstein farmers on the dairy herd management computer program DairyComp (version 22.6.0, Valley Agricultural Software) from 2007 to 2019 were available from CanWest DHI (a member of Lactanet) and included information from 781,583 cows within 1,192 herds. After removing herds where BPD were not available, final data contained records from 707,240 cows within 981 herds, which represented 73.15% of cows from the national Canadian Holstein herd on July 1, 2020 (Statistics Canada, 2020).

A total of 5,804 BPD was available, with frequencies ranging from 1 to 24 BPD/herd (mean = 5.8 BPD/herd). From these, 2,481 BPD were unique and used

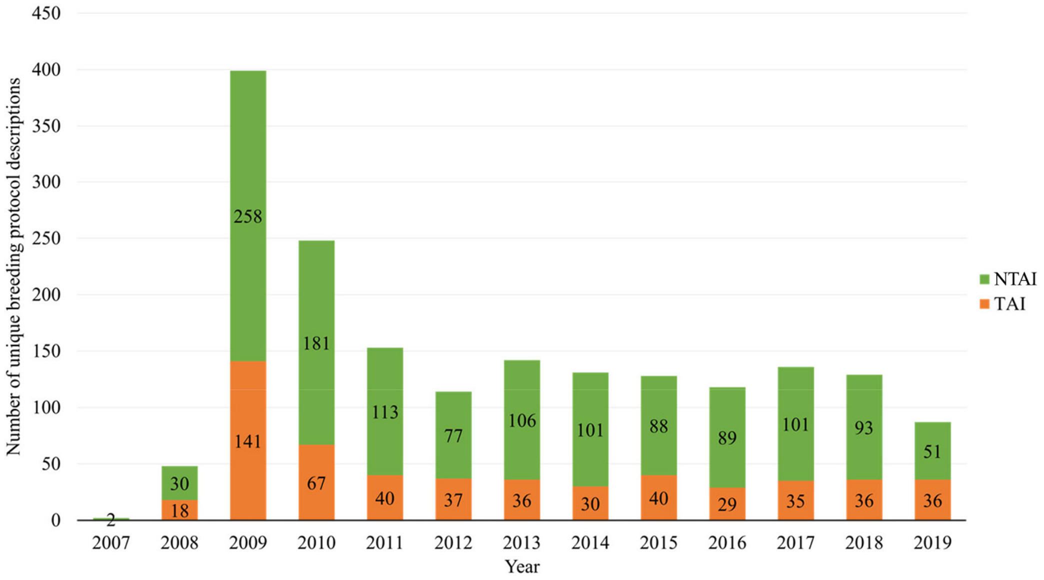


Figure 1. Number of unique breeding protocol descriptions (BPD) used for the first time from 2007 to 2019 among all herds. Numbers within bars represent the number of records per group in each year (e.g., in 2011, there were a total of 40 unique breeding protocol descriptions used for timed artificial insemination (TAI) protocols that were not previously used from 2007–2010]. Orange: BPD containing TAI protocols; green: BPD that do not contain TAI protocols (NTAI).

in at least 1 herd. Sixty-six percent of herds (n = 650) had 1 or more herd-specific BPD, totaling 2,021 BPD not shared among herds. The remaining 460 BPD were shared by 2 or more herds. The most frequent BPD were "Estrumate," "CIDR," "Ovsynch," and "Natural," which were used in 101, 166, 224, and 252 herds, respectively. Herd-specific BPD included "15HRS," "2xBred-SH," and "GGPG+Cidr."

Although many BPD (2,410) were used in less than 10 herds, the majority were variations of a given description (e.g., CIDR-Ovsynch, CIDR-SYNCH, CIDR/PRID), and others contained typos or combinations of lower- and uppercase characters (e.g., CIDR-Sync, CIDR-synch, Cidr-synch). These similarities were expected to help in the model learning process by reducing feature dimensionality. Even though other BPD seemed to be less meaningful (e.g., ttt, rpt, cold, WB, CM, and Reg-2), or could be related to treatment of sick animals (e.g., CIDR for treatment of cystic ovaries), they remained in the dataset as they added important features to help the models understand what a TAI BPD does not look like, thus increasing overall precision and recall.

Breeding protocol descriptions were grouped into the following 2 main classes: (1) TAI, when TAI protocol was used alone (e.g., "Ovsynch") or in combination with any other protocol (e.g., "PRID & Estrumate" or "Ovsynch & Natural"); and (2) NTAI, when BPD did not indicate any use of TAI protocols, such as heat detection (e.g., "standing heat"), hormone use (e.g., "Estrumate"), or unclear descriptions (e.g., "vet advice"). It was assumed that classes were previously labeled with high accuracy by Lynch et al. (2021).

Data Preprocessing

Transformation of all BPD to lowercase and the replacement of underscores to single spaces were performed, leading to a reduction of 26.0% in the number of unique BPD, from 2,481 to 1,835. The number of new unique BPD (Figure 1) added to the dataset every year stabilized after 2010; therefore, data from 2007 to 2010 were pooled as 1 group. From 2010 to 2019, an average of 139 new unique BPD was used yearly, which represents an average increase of 15.8% per year.

Feature Construction for Training and Test Datasets

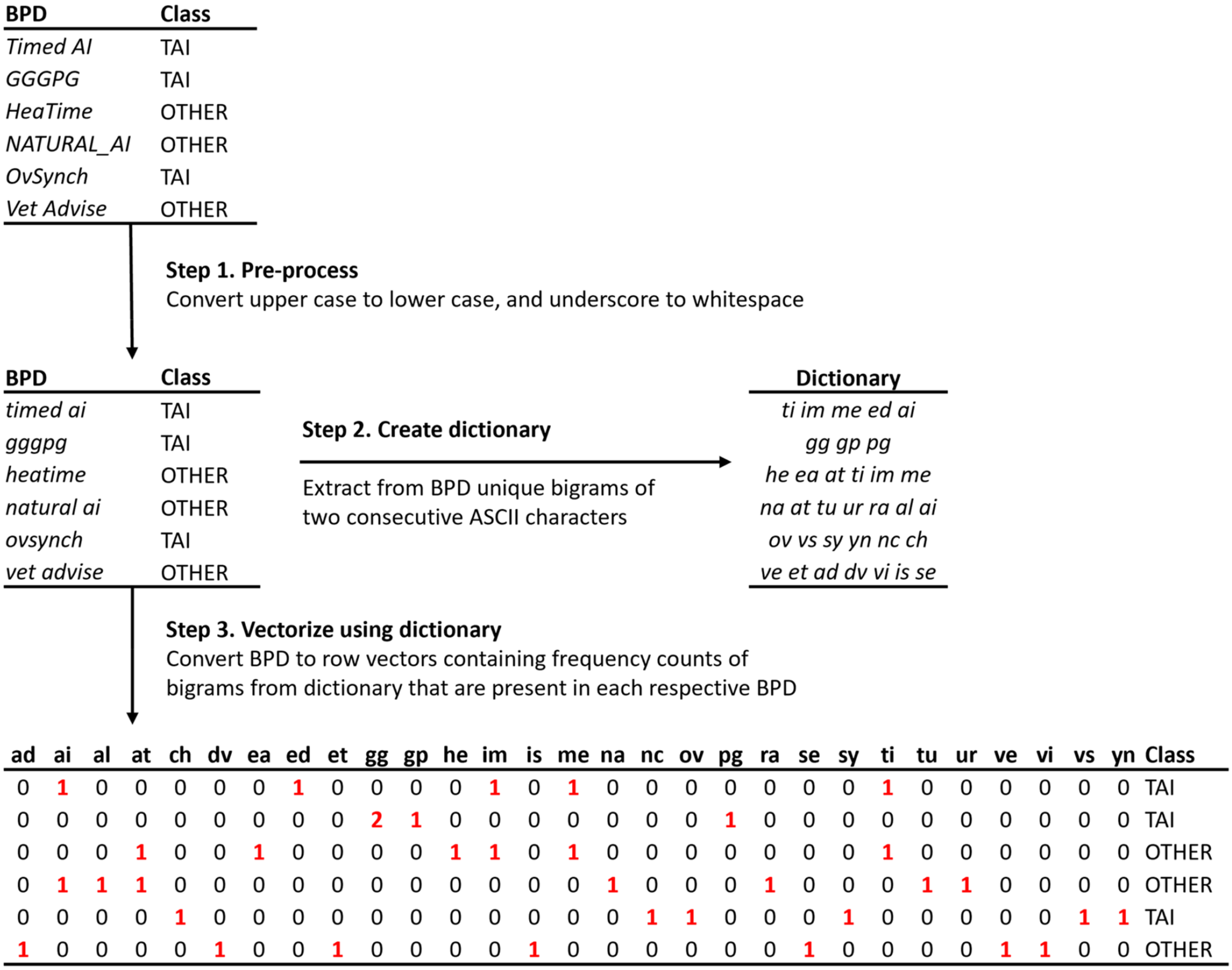
Nine training datasets were created by grouping BPD according to years of use starting from 2007 to 2010 and incrementally adding 1 yr to the previous group until 2018 (i.e., 2007–2010, 2007–2011, 2007–2012, ..., 2007–2018).

Construction of new features (Figure 2) for the learning problem was performed using a character bigram approach with the *CountVectorizer()* function from the Scikit-learn software, version 0.24.2 (Pedregosa et al., 2011), implemented in Python 3.8. This approach was assumed to reduce the eventual differences between the official Canadian languages (i.e., French and English).

A dictionary specific to each training set was created with all possible unique bigrams of 2 consecutive ASCII characters present among all BPD in a training set. Bigrams formed with a whitespace (i.e., representing start- and end-of-word letters) and duplicated bigrams were removed from the dictionary. The dictionary was used in a vectorizer function to convert BPD into a sparse matrix of character bigram counts of size n by m, where n represents the number of unique BPD, and m is the size of the dictionary (number of character bigrams). Feature construction created between 491

Table 1. Number of records and constructed features in each training dataset

Training dataset	Number of records	Number of features
2007–2010	697	491
2007–2011	850	540
2007–2012	964	580
2007–2013	1,106	611
2007–2014	1,237	636
2007–2015	1,365	653
2007–2016	1,483	676
2007–2017	1,619	704
2007–2018	1,748	727



BPD	Class
Timed AI	TAI
GGGPG	TAI
HeaTime	OTHER
NATURAL_AI	OTHER
OvSynch	TAI
Vet Advise	OTHER

Step 1. Pre-process
Convert upper case to lower case, and underscore to whitespace

BPD	Class
timed ai	TAI
gggpg	TAI
heatime	OTHER
natural ai	OTHER
ovsynch	TAI
vet advise	OTHER

Step 2. Create dictionary
Extract from BPD unique bigrams of two consecutive ASCII characters

Dictionary
ti im me ed ai
gg gp pg
he ea at ti im me
na at tu ur ra al ai
ov vs sy yn nc ch
ve et ad dv vi is se

Step 3. Vectorize using dictionary
Convert BPD to row vectors containing frequency counts of bigrams from dictionary that are present in each respective BPD

ad	ai	al	at	ch	dv	ea	ed	et	gg	gp	he	im	is	me	na	nc	ov	pg	ra	se	sy	ti	tu	ur	ve	vi	vs	yn	Class
0	1	0	0	0	0	0	1	0	0	0	0	1	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	TAI
0	0	0	0	0	0	0	0	0	2	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	TAI
0	0	0	1	0	0	1	0	0	0	0	1	1	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	OTHER
0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	1	1	0	0	0	0	OTHER
0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	1	0	0	0	0	0	1	1	TAI
1	0	0	0	0	1	0	0	1	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	1	1	0	0	OTHER

Figure 2. Example of data preprocessing and feature construction. Step 1: breeding protocol descriptions (BPD) were preprocessed by converting uppercase to lowercase, and underscore to whitespace. Step 2: a dictionary specific to each dataset was created with all possible unique bigrams of 2 consecutive ASCII characters present among all BPD in the dataset. In the example above, this dictionary contains 29 unique character bigrams. Step 3: this dictionary was used in a vectorizer function to convert BPD into a sparse matrix with row vectors of frequency counts of character bigrams from the dictionary that were present in the BPD. This matrix had the size n by m, where n represents the number of unique BPD, and m the size of the dictionary (number of character bigrams). Each column vector in the matrix represents one feature. TAI = timed artificial insemination protocols.

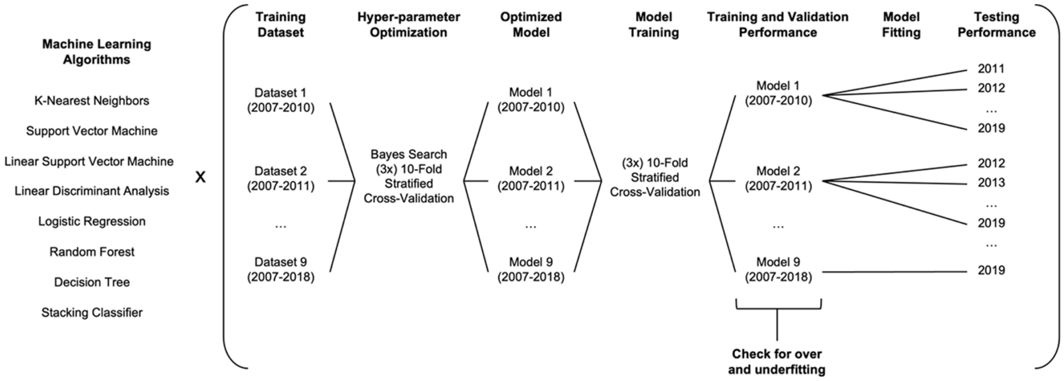


Figure 3. Overview of the modeling pipeline used in this study. Eight machine learning algorithms were used to build classification models, and a Bayesian search approach was applied to optimize their hyperparameters according to different training datasets. A total of 9 training datasets were used, each increasing the amount of data incrementally every year. To minimize overfitting, optimized models were trained with a stratified 10-fold cross-validation approach. Training and validation scores were used to detect overfitting and rank algorithms based on performance of their models on validation data across all training datasets. Testing of models were performed on unseen data of the year(s) following the range of years used for their respective training datasets.

(2007–2010) and 727 (2007–2018) unique character bigrams, with an average increase of 4.5% new features per year (Table 1).

A total of 45 test datasets were created, each consisting of unique BPD used in the years following a given training dataset until 2019, using 1 file per year. For example, models trained with data from 2007 to 2010 were tested on 9 datasets (i.e., 1 for each year from 2011–2019), whereas models trained with data from 2007 to 2017 were only tested on 2 datasets (i.e., 1 with unique BPD used in 2018, and another in 2019). Test datasets were further adjusted so that all and only those bigrams present in the corresponding training dataset used to train the model were kept, whereas the rest were discarded. Furthermore, for a fair comparison of testing performance, each test dataset consisted of a defined number of 87 randomly sampled records, which was the number of records available in the smallest test dataset (i.e., in 2019).

Modeling Pipeline

Algorithms. Supervised ML classification algorithms were evaluated for the classification of BPD used with Canadian Holsteins: decision tree classifier (**DT**), k-nearest neighbors classifier (**KNN**), linear discriminant analysis (**LDA**), logistic regression (**LR**), support vector machines classifier (**SVM**), linear SVM (**LSVM**), random forest (**RF**), and stacking classifier (**STK**). Stacking classifier is an ensemble algorithm that works with 2 levels of models. Level zero contains already trained base models, whereas level 1 uses a meta-model that is responsible to learn how to best combine the predictions given by the base models by deducing the biases of the generalizers from level zero in a second space. The meta-model uses as its inputs the predictions of the base models taught with part of the learning set and tries to predict the rest of it (Wolpert, 1992). In this study, models produced with the former 7 algorithms were used as base models, whereas LR was also used as the algorithm for the meta-model. Therefore, a total of 8 algorithms were considered, using appropriate functions from Scikit-learn, and a diagram with the modeling pipeline is shown in Figure 3.

Performance Measures. In a classification problem, a confusion matrix is typically created representing the summary of the number of correct and incorrect prediction results broken down by each class. There has been a long discussion around performance measures used to evaluate classification problems based on the confusion matrix, and no widespread consensus has been reached on a single measure yet (Brown, 2018). Accuracy is a very popular statistical measure; however, it can show overoptimistic inflated results, especially when imbalanced datasets are analyzed (Chicco and Jurman, 2020).

Alternatively, the Matthews correlation coefficient (**MCC**) is a more reliable statistical measure for imbalanced datasets (Matthews, 1975). The MCC produces a high score only if the prediction obtained good results in all 4 categories of the confusion matrix and it is proportional to the size of positive and negative elements in the dataset, as shown in Equation 1 as follows:

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}, \quad [1]$$

where TP, TN, FP, and FN are true positive, true negative, false positive, and false negative values, respectively. Although MCC and accuracy include all 4 result types in their formulations, only MCC includes both type-I (FP) error and type-II (FN) error in its numerator, and in a multiplicative manner, which penalizes both types (Brown, 2018).

Therefore, MCC was used as the main statistic to measure model performance, and it was calculated using the *matthews_corrcoef()* function from the Scikit-learn library in Python. Nonetheless, due to its mathematical properties, extreme optimization on either positives (TP, FP) or negatives (FP, FN) can yield MCC values close to zero, thus both positive and negative error rates must be low to achieve a high MCC (Brown, 2018). To account for extreme optimization scenarios, precision and recall were used as complementary measures to carefully interpret MCC scores, as well as consideration of values from the confusion matrix. The functions *precision_score()*, *recall_score()*, and *f1_score()* from Scikit-learn were used, respectively, having TAI as the positive label. We provide below the formulae for the 2 considered measures.

Precision (positive predictive value) is the ratio of the number of true positive examples out of those that were classified as positive (Equation 2).

$$Precision = \frac{TP}{TP + FP}. \qquad [2]$$

Recall (sensitivity or true positive rate) is the ratio of examples correctly predicted as positive to the number of actual positive examples (Equation 3).

$$Recall = \frac{TP}{TP + FN}. \qquad [3]$$

The aim of this study was to correctly identify TAI protocols so that their effects on fertility could be properly managed during genetic evaluations for fertility; therefore, an algorithm that produces models that maximize both precision and recall is more desirable. For this reason, algorithms were ranked based on the average F1-score (Equation 4) of their respective models. The F1-score is a weighted (harmonic) average of the precision and recall scores, with values ranging from 0 (lowest) to 1 (highest). Differences in F1 scores across algorithms and models were tested via mixed model ANOVA in SAS PROC GLM. The ANOVA model included test sets nested in models as a categorical random factor, and algorithms and models as fixed categorical factors. Scheffé adjustment for multiple comparisons was used to control type I error rate.

$$F1 - Score = \frac{2 \times Precision \times Recall}{Precision + Recall}. \qquad [4]$$

Optimization of Model Hyperparameters. A model hyperparameter is an external user-tunned parameter whose value is used to control various aspects of the learning process such as speed and quality, tree depth in DT models, or number of neighbors (k) in KNN models, and it cannot be inferred when fitting the model to the data. In this study, optimization of hyperparameters was performed with a cross-validated search over various hyperparameter settings using a Bayesian approach with the *BayesSearchCV()* function from the Scikit-optimize module, version 0.8.1 (Head et al., 2020), implemented in Python 3.8. The full list of algorithms and their respective optimized model hyperparameters are given in Table 2. The set of hyperparameters that produced maximum MCC scores in the validation (hold-out) data was ultimately selected for any given model. All models had their hyperparameters optimized based on the training set, except for the ones using the STK algorithm because they were built based on the models optimized here. Therefore, 63 models were optimized in total (7 algorithms × 9 training sets).

Model Training and Testing. Models were trained using a stratified 10-fold cross-validation approach repeated 3 times, as suggested for classification problems (Kohavi, 1995), using the *RepeatedStratifiedKFold()* function implemented in Scikit-learn. Briefly, each training set was split in 10 parts with a similar distribution of records per class (TAI: 70%; NTAI: 30%). One part was set aside (hold-out) for validation purposes, whereas the other 9 were used to train the model and measure its performance. This process was performed 10 times until all parts had their turn to be used for validation. Performances on the validation parts were averaged and are reported herein as validation scores, whereas the average performances of the model on the other parts used to train the model are referred as training scores.

All model hyperparameters were optimized based on different training sets; therefore, models that used the same algorithm did not necessarily have the same hyperparameter values. Therefore, we ranked the performance of models on the same training data and averaged such rankings according to the algorithm used. The algorithm whose models had the best average MCC ranking on the validation data across all training sets was considered the best model.

To precisely gauge the ability of the alternative algorithms to produce models able to classify future data, models were tested on unseen datasets of the same size. Test datasets were constructed with a standard-

Table 2. List of classification algorithms and values used for hyper-parameter optimization; hyperparameter names and values are shown as available from their respective functions in the Scikit-learn software, version 0.24.2, implemented in Python 3.8

Algorithm[1]	Hyperparameters	Values
LR	class_weight	balanced, none
	solver	newton-cg, lbfgs, liblinear, sag, saga
	C	Range from 0.1 to 2.0 in increments of 0.1
KNN	n_neighbors	Range from 1 to 11 in increments of 2
	leaf_size	Range from 10 to 50 in increments of 10
	metric	euclidean, manhattan, chebyshev, minkowski, hamming, braycurtis
DT	criterion	gini, entropy
	max_features	auto, sqrt, log2, none
	splitter	best, random
	max_depth	Range from 2 to 26 in increments of 1, none
SVM	C	Range from 0.01 to 2.0 in increments of 0.2
	kernel	poly, rbf, sigmoid
	class_weight	balanced, None
	degree	Range from 1 to 3, in increments of 1
LSVM	C	Range from 0.01 to 2.0 in increments of 0.2
	loss	hinge, squared_hinge
	class_weight	balanced, None
LDA	solver	lsqr, svd
RF	max_samples	Range from 0.1 to 0.9 in increments of 0.1, none
	max_depth	Range from 1 to 26 in increments of 1, none
	criterion	gini, entropy
	max_features	Range from 1 to number of features in increments of 50
	n_estimators	Range from 1 to 1,000 in increments of 100

[1] LR: logistic regression; KNN: k-nearest neighbors classifier; DT: decision tree classifier; SVM: support vector machines classifier; LSVM: linear SVM; LDA: linear discriminant analysis; RF: random forest.

ized number of 87 observations, which were extracted from the years following the ones used in their respective training set. For example, a model optimized and trained with data from 2007 to 2012 was tested on 7 different testing sets (2013, 2014, [. . .], and 2019), whereas a model optimized and trained with data from 2007 to 2018 was only tested on data from 2019.

RESULTS AND DISCUSSION

Model Training

Models were optimized for each training set using a cross-validated Bayesian search approach (Head et al., 2020). The combination of hyperparameters that resulted in maximum MCC in the hold-out data was chosen as the best one for each model according to the training set used. Optimized models were trained on a range of training sets and their respective performances with averaged cross-validated training, and validation scores are shown in Figure 4.

Six algorithms (DT, LR, SVM, LSVM, RF, and STK) produced models that performed extremely well on all training sets, with an average training score of 0.99 ± 0.00 (mean ± SD; Table 3) and a validation score of 0.95 ± 0.01 on the hold-out data (Table 4). Models using the STK algorithm performed the best (highest MCC rank on all held-out data of all training sets), with the lowest MCC of 0.96 ± 0.02 (2007–2010) and highest of 0.97 ± 0.04 (2007–2015; Table 4). On the other hand, LDA models had the lowest performance (lowest MCC rank on all held-out data of all training sets), with the lowest MCC of 0.73 ± 0.08 (2007–2010) and highest of 0.87 ± 0.04 (2007–2017; Table 4).

We also investigated if any of the models were affected by over- and underfitting. A good machine learning model aims to generalize well from the training data so it can make accurate predictions on unseen data. However, overfitting happens when a model learns or "memorizes" the details and noise in the training data to the extent that it negatively affects the predictive performance of the model on new data. In contrast, an underfitted model fails to adequately capture the relationships between the variables in the data due to its simplicity (e.g., insufficient number of features). The results depicted in Figure 4 suggest that LDA was overfitted because MCC scores on training data were consistently high (0.98 ± 0.01), whereas those for validation data were lower (0.82 ± 0.05). The LDA's limitation to generalize new data can also be noticed by a plateau in validation scores from 2015 to 2018. In contrast, KNN models exhibited a slight underfitting because their training scores decreased from 0.93 to 0.90 with addition of more data over the years until 2015, whereas validation scores varied slightly in the

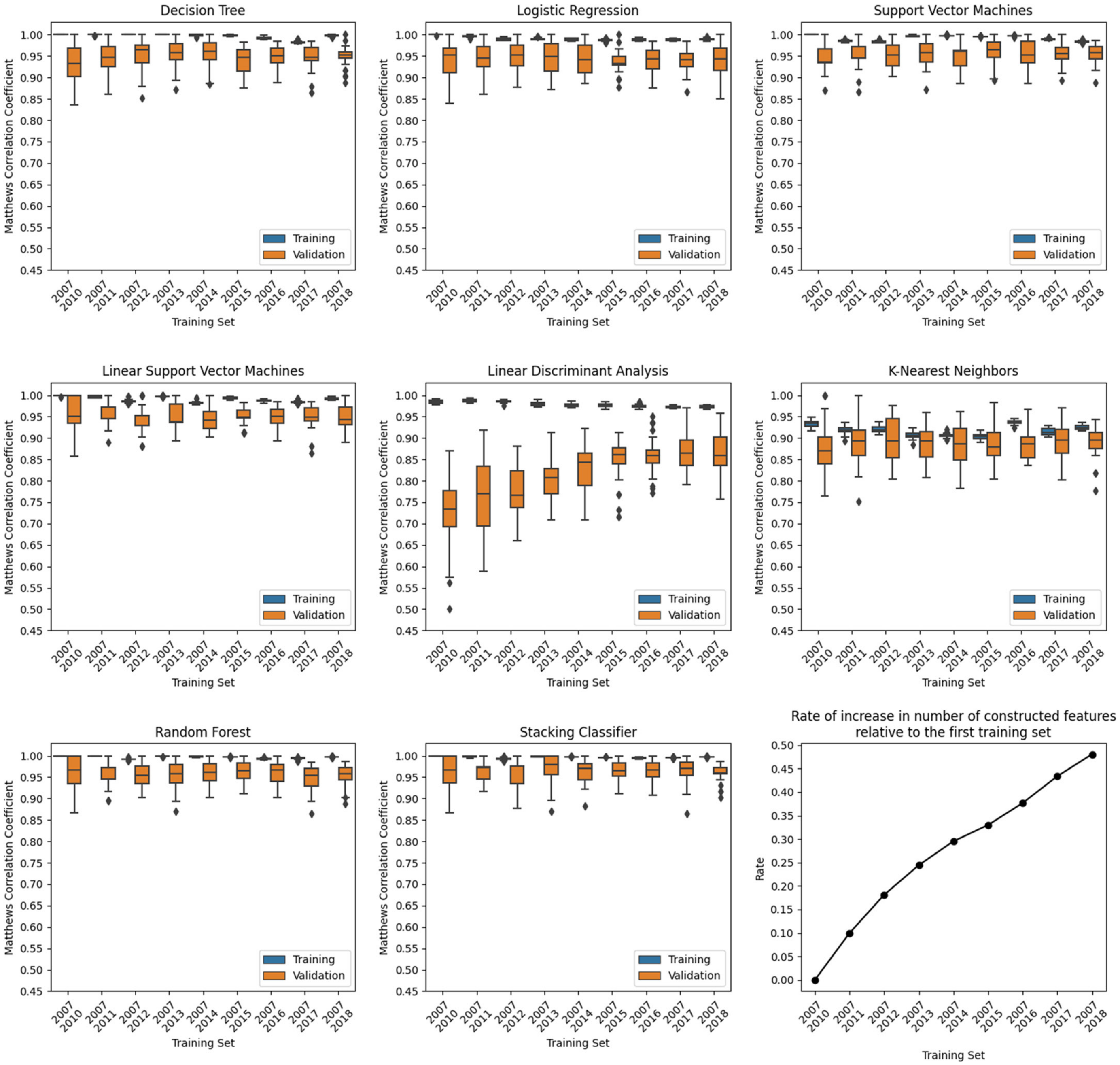


Figure 4. Each box-plot displays the 5-number summary of either the training (blue) or validation (orange) scores (Matthews correlation coefficient) from a 3×-repeated stratified 10-fold cross-validation of estimators with hyperparameters optimized for each training set using a Bayesian search approach. The 5-number summary is the minimum (lower end of vertical line), first quartile (bottom of the box), median (line inside box), third quartile (top of the box), and maximum (upper end of vertical line). Dots on the first 8 plots (left to right, top to bottom) represent outliers. Bottom right line plot shows the rate of increase in number of constructed features relative to the first training set.

same period, from 0.88 to 0.89. Even though training performance increased with addition of training data in 2016, validation scores remained constant and KNN was still outperformed by all the other models, except LDA.

Model Testing

Over the past decade, at least 87 new codes were used by farmers every year to describe breeding protocols for Canadian Holsteins (Figure 1). Therefore, we tested the

Table 3. Average training scores (Matthews correlation coefficient) for optimized models[1] obtained from a stratified 10-fold cross-validation using different training datasets; SD varied from 0 to 0.01

Training dataset	DT	KNN	LDA	LR	LSVM	RF	SVM	STK
2007–2010	1.00	0.93	0.99	1.00	1.00	1.00	1.00	1.00
2007–2011	1.00	0.92	0.99	1.00	1.00	1.00	0.99	1.00
2007–2012	1.00	0.92	0.99	0.99	0.99	0.99	0.98	0.99
2007–2013	1.00	0.91	0.98	0.99	1.00	1.00	1.00	1.00
2007–2014	1.00	0.91	0.98	0.99	0.98	1.00	1.00	1.00
2007–2015	1.00	0.90	0.98	0.99	0.99	1.00	1.00	1.00
2007–2016	0.99	0.94	0.98	0.99	0.99	0.99	1.00	1.00
2007–2017	0.98	0.92	0.97	0.99	0.99	0.99	0.99	1.00
2007–2018	1.00	0.93	0.97	0.99	0.99	1.00	0.98	1.00

[1] LR: logistic regression; KNN: k-nearest neighbors classifier; DT: decision tree classifier; SVM: support vector machines classifier; LSVM: linear SVM; LDA: linear discriminant analysis; RF: random forest; STK: stacking classifier.

ability of the models to predict randomly selected (n = 87) unseen BPD from 2011 to 2019, according to their respective training dataset.

Models using the STK algorithm outperformed all the other models when tested on unseen data and had similar MCC scores on most testing data (0.94 ± 0.04; Table 5). On average, breeding codes used in 2018 were the easiest to predict (0.99 ± 0.01) by STK models, whereas those from 2013 posed a bigger challenge (0.89 ± 0.02). Among STK models, the model trained with BPD used from 2007 to 2010 had the lowest average performance on the testing datasets (MCC = 0.92 ± 0.05), whereas the one trained with data from 2007 to 2018 that predicted BPD used in 2019 had an MCC performance of 0.98.

For this learning problem, a precision equal to 1 means that all TAI observations were correctly classified as TAI. On the extreme opposite, if all TAI observations were wrongly classified as NTAI, precision would be equal to zero. If all observations classified as TAI were truly TAI, recall would be equal to 1, regardless of how many NTAI labels were misclassified as TAI. However, if none of the true TAI were classified as TAI, recall would be equal to zero, no matter how many BPD were correctly classified as NTAI.

An algorithm that produces a model that maximizes both precision and recall is more desirable; therefore, algorithms were ranked based on the average F1-score of their respective models. On average, models using the STK algorithm had the highest F1-score (0.96 ± 0.03), followed by DT (0.95 ± 0.04), RF (0.95 ± 0.05), LSVM (0.94 ± 0.04), SVM (0.94 ± 0.04), LR (0.93 ± 0.05), KNN (0.89 ± 0.04), and LDA (0.82 ± 0.08; Table 6).

Analysis of variance results (Supplemental Data S1, https://data.mendeley.com/datasets/ptmgr4vcz7/1; Alacantra, 2022) indicated that there were no significant differences (P-value > 0.15) between F1-scores from all the models, except for KNN and LDA models, which performed significantly worse than all the other models (P-value <0.0001). Therefore, DT, LR, SVM, LSVM, RF, and STK algorithms seemed to produce models that can predict new BPD with very similar performance.

In general, incremental addition of new data to the training dataset every year did not help the maximiza-

Table 4. Average validation scores (Matthews correlation coefficient) for optimized models[1] obtained from a stratified 10-fold cross-validation using different training datasets; SD varied from 0.02 to 0.09

Validation dataset	DT	KNN	LDA	LR	LSVM	RF	SVM	STK
2007–2010	0.93	0.88	0.73	0.95	0.95	0.96	0.95	0.96
2007–2011	0.95	0.89	0.77	0.95	0.95	0.96	0.95	0.96
2007–2012	0.95	0.90	0.78	0.94	0.95	0.96	0.96	0.96
2007–2013	0.95	0.89	0.80	0.94	0.95	0.96	0.96	0.97
2007–2014	0.95	0.88	0.83	0.94	0.94	0.96	0.95	0.97
2007–2015	0.94	0.89	0.85	0.94	0.95	0.96	0.96	0.97
2007–2016	0.95	0.89	0.86	0.94	0.95	0.96	0.96	0.97
2007–2017	0.95	0.89	0.87	0.94	0.95	0.95	0.96	0.97
2007–2018	0.95	0.89	0.86	0.94	0.95	0.95	0.95	0.96

[1] LR: logistic regression; KNN: k-nearest neighbors classifier; DT: decision tree classifier; SVM: support vector machines classifier; LSVM: linear SVM; LDA: linear discriminant analysis; RF: random forest; STK: stacking classifier.

Table 5. Average Matthews correlation coefficient of models[1] on test sets from 2011 to 2019 according to their machine learning algorithm; models predicted unseen breeding protocol descriptions used in each year that followed the date range (2007–2010 to 2007–2019) used in their respective training dataset[2]

Test dataset	DT	KNN	LDA	LR	LSVM	RF	SVM	STK
2011[3]	0.98	0.95	0.66	0.95	0.95	0.98	0.95	0.95
2012	0.91	0.90	0.70	0.87	0.90	0.92	0.92	0.92
2013	0.83	0.78	0.60	0.82	0.80	0.82	0.85	0.89
2014	0.96	0.88	0.65	0.88	0.93	0.96	0.90	0.93
2015	0.93	0.90	0.79	0.97	0.96	0.96	0.94	0.97
2016	0.90	0.79	0.73	0.82	0.86	0.87	0.92	0.92
2017	0.87	0.82	0.78	0.94	0.95	0.88	0.87	0.92
2018	0.99	0.88	0.80	0.97	0.94	0.99	0.98	0.99
2019	0.98	0.86	0.85	0.92	0.93	0.96	0.93	0.96

[1] LR: logistic regression; KNN: k-nearest neighbors classifier; DT: decision tree classifier; SVM: support vector machines classifier; LSVM: linear SVM; LDA: linear discriminant analysis; RF: random forest; STK: stacking classifier.

[2] All test sets had the same number (n = 87) of random observations. SD varied from 0 to 0.1.

[3] Not an average; results from models trained only with data from 2007 to 2010.

tion of F1-scores for all models (Figure 5). Analysis of variance results (Supplemental Data S1) showed no significant differences in the F1-scores from all the models by adding data to the training set every year (P-value = 0.43). Such results showed that the ML algorithms used in this study were robust and could be used to produce models with consistently high performance regardless of the amount of data used for training. In practice, this means that models produced with such algorithms would not need to be optimized and trained before every genetic evaluation.

Applications

Breeding data are recorded differently by farmers, thus there is no "one-fits-all" type of model that can take a text-based description of a breeding protocol and classify it as either TAI or not TAI. Therefore, in this study we provided the dairy industry with a methodology that is flexible enough to vectorize any text-based BPD and develop classification models aiming to identify TAI protocols among NTAI breeding protocols using machine learning algorithms. Our main concern was to provide algorithms that were robust enough to produce models that provide high performance on test data regardless of the amount of data used for training. This is especially important because it reduces the need of frequent retraining, otherwise required due to newly created BPD by farmers yearly. Such methodology can be used by researchers to further understand the effects of TAI on fertility evaluations and for improvements on genetic evaluations to account for the effects of TAI within existing automated systems.

Table 6. Average F1-score of models[1] on test sets from 2011 to 2019 according to their machine learning algorithm[2]

Test dataset	DT	KNN	LDA	LR	LSVM	RF	SVM	STK
2011[3]	0.98	0.97	0.78	0.97	0.97	0.98	0.97	0.97
2012	0.94	0.93	0.80	0.92	0.93	0.95	0.95	0.95
2013	0.87	0.84	0.71	0.87	0.86	0.87	0.89	0.92
2014	0.97	0.90	0.71	0.90	0.94	0.97	0.91	0.94
2015	0.95	0.92	0.85	0.98	0.97	0.97	0.95	0.98
2016	0.92	0.82	0.79	0.85	0.89	0.89	0.93	0.93
2017	0.90	0.86	0.83	0.95	0.96	0.90	0.90	0.94
2018	1.00	0.91	0.86	0.98	0.96	1.00	0.99	0.99
2019	0.99	0.91	0.91	0.95	0.96	0.97	0.96	0.97

[1] LR: logistic regression; KNN: k-nearest neighbors classifier; DT: decision tree classifier; SVM: support vector machines classifier; LSVM: linear SVM; LDA: linear discriminant analysis; RF: random forest; STK: stacking classifier.

[2] Models predicted unseen breeding protocol descriptions used in each year that followed the date range (2007–2010 to 2007–2019) used in their respective training dataset. All test sets had the same number (n = 87) of random observations. SD varied from 0 to 0.08.

[3] Not an average; results from models trained only with data from 2007 to 2010.

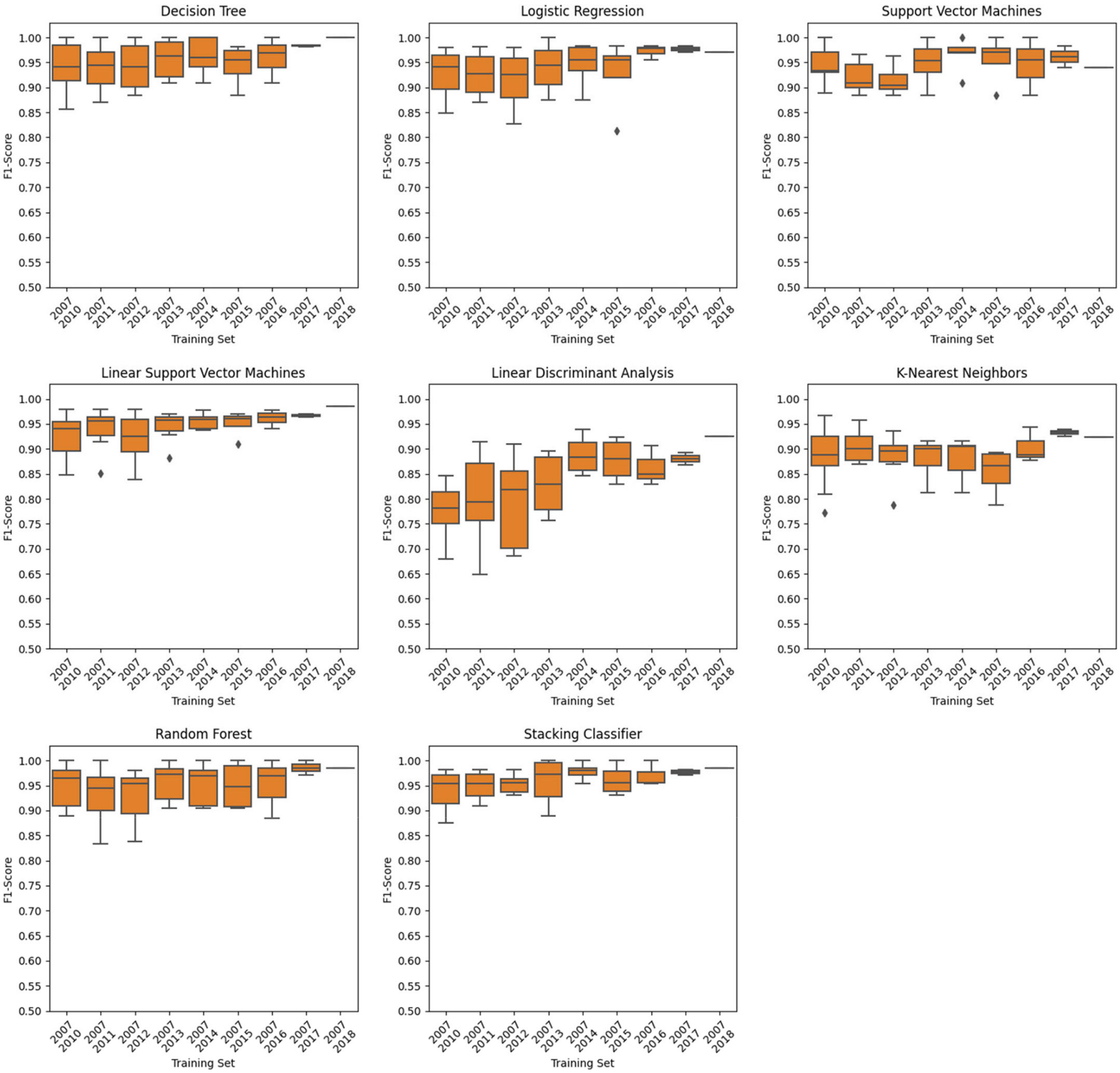


Figure 5. Distribution of F1-scores of models on test sets according to their machine learning algorithm and training set. Models predicted unseen breeding protocol descriptions used in each year that followed the date range used in their respective training dataset (e.g., the first boxplot of each graph contains the distribution of results from test sets from 2011–2019 from a model trained with data from 2007–2010, whereas the last boxplot is the result from the test set from 2019 from a model trained with data from 2007–2018). All test sets had the same number (n = 87) of randomly selected observations. Each box plot displays the 5-number summary of the F1-scores of each test set, which are the minimum (lower end of vertical line), first quartile (bottom of the box), median (line inside box), third quartile (top of the box), and maximum (upper end of vertical line); dots represent outliers.

CONCLUSIONS

In this study, we used 8 supervised machine learning algorithms to classify BPD with constructed features representing the combination of 2 consecutive ASCII characters. Our results showed that models that used the STK algorithm (i.e., a stacking classifier that included DT, LDA, KNN, SVM, LSVM, LR, and RF) showed higher performance in the prediction of unseen

BPD used in each year that followed the years used in their respective training datasets (2007–2010 to 2007–2019). We also showed precision and recall were maximized by STK models, but their F1-scores on test data were not different from DT, SVM, LSVM, LR, and RF. Altogether, results presented herein suggest machine learning algorithms can be used to produce robust models that correctly identify TAI protocols from dairy cattle breeding records, thus opening the possibility for unbiased genetic evaluation of animals based on their natural fertility.

ACKNOWLEDGMENTS

This research was financially supported by Agriculture and Agri-Food Canada (Ottawa, Canada) and by additional contributions from Dairy Farmers of Canada (Ottawa, Canada), Lactanet (Guelph, Canada), and the Canadian Dairy Commission (Ottawa, Canada) under the Agri-Science Clusters Initiative. As per the research agreement, aside from providing financial support, the funders have no role in the design and conduct of the studies, data collection and analysis or interpretation of the data. Researchers maintain independence in conducting their studies, own their data, and report the outcomes regardless of the results. The decision to publish the findings rests solely with the researchers. This research is also part of the activities of C. F. Baes, D. Tulpan and F.S. Schenkel within the Resilient Dairy Genome Project (Guelph, Canada; http://www.resilientdairy.ca). C. F. Baes acknowledges support from the Natural Sciences and Engineering Research Council of Canada, Canada Research Chair program. The authors have not stated any conflicts of interest.

REFERENCES

Aiken, V. C. F., J. R. R. Dórea, J. S. Acedo, F. G. de Sousa, F. G. Dias, and G. J. M. Rosa. 2019. Record linkage for farm-level data analytics: Comparison of deterministic, stochastic and machine learning methods. Comput. Electron. Agric. 163:104857. https://doi.org/10.1016/j.compag.2019.104857.

Alcantara, L. 2022. ANOVA Results, Mendeley Data, V1. https://doi.org/10.17632/ptmgr4vcz7.1.

Brown, J. B. 2018. Classifiers and their metrics quantified. Mol. Inform. 37:1700127. https://doi.org/10.1002/minf.201700127.

Cerri, R. L. A., J. E. P. Santos, S. O. Juchem, K. N. Galvão, and R. C. Chebel. 2004. Timed artificial insemination with estradiol cypionate or insemination at estrus in high-producing dairy cows. J. Dairy Sci. 87:3704–3715. https://doi.org/10.3168/jds.S0022-0302(04)73509-2.

Chicco, D., and G. Jurman. 2020. The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. BMC Genomics 21:6. https://doi.org/10.1186/s12864-019-6413-7.

Denis-Robichaud, J., R. L. A. Cerri, A. Jones-Bitton, and S. J. LeBlanc. 2016. Survey of reproduction management on Canadian dairy farms. J. Dairy Sci. 99:9339–9351. https://doi.org/10.3168/jds.2016-11445.

HaCohen-Kerner, Y., D. Miller, and Y. Yigal. 2020. The influence of preprocessing on text classification using a bag-of-words representation. PLoS One 15:e0232525. https://doi.org/10.1371/journal.pone.0232525.

Haldar, R., and D. Mukhopadhyay. 2011. Levenshtein distance technique in dictionary lookup methods: An improved approach. Comput. Inf. Sci. https://doi.org/https://doi.org/10.48550/arXiv.1101.1232.

Head, T., M. Kumar, H. Nahrstaedt, G. Louppe, and I. Shcherbatyi. 2020. Scikit-optimize: Sequential model-based optimization in Python. doi:https://doi.org/10.5281/zenodo.4014775.

Kohavi, R. 1995. A study of cross-validation and bootstrap for accuracy estimation and model selection. Pages 1137–1143 in Proceedings of the 14th International Joint Conference on Artificial Intelligence, Montreal, Canada. Morgan Kaufmann Publishers Inc.

Lactanet. 2022a. Company History & Mandate. Accessed May 11, 2022. https://www.cdn.ca/company.php.

Lactanet. 2022b. Insemination File Layout. Accessed May 11, 2022. https://www.cdn.ca/download.php?/layouts/DESinsemination.pdf.

Lecluze, C., L. Rigouste, E. Giguet, and N. Lucas. 2013. Which granularity to bootstrap a multilingual method of document alignment: Character N-grams or word N-grams? Procedia Soc. Behav. Sci. 95:473–481. https://doi.org/10.1016/j.sbspro.2013.10.671.

Lynch, C., G. A. Oliveira Junior., F. S. Schenkel, and C. F. Baes. 2021. Effect of synchronized breeding on genetic evaluations of fertility traits in dairy cattle. J. Dairy Sci. 104:11820–11831. https://doi.org/10.3168/jds.2021-20495.

Matthews, B. W. 1975. Comparison of the predicted and observed secondary structure of T4 phage lysozyme. Biochim. Biophys. Acta. 405:442–451. https://doi.org/10.1016/0005-2795(75)90109-9.

Oliveira Junior, G. A., L. R. Schaeffer, F. Schenkel, F. Tiezzi, and C. F. Baes. 2021. Potential effects of hormonal synchronized breeding on genetic evaluations of fertility traits in dairy cattle: A simulation study. J. Dairy Sci. 104:4404–4412. https://doi.org/10.3168/jds.2020-18944.

Pedregosa, F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and É. Duchesnay. 2011. Scikit-learn: Machine learning in Python. J. Mach. Learn. Res. 12:2825–2830.

Ribeiro, E. S., K. N. Galvão, W. W. Thatcher, and J. E. P. Santos. 2012. Economic aspects of applying reproductive technologies to dairy herds. Anim. Reprod. Sci. 9:370–387.

Roelofs, J., F. López-Gatius, R. H. F. Hunter, F. J. C. M. van Eerdenburg, and Ch. Hanzen. 2010. When is a cow in estrus? Clinical and practical aspects. Theriogenology 74:327–344. https://doi.org/10.1016/j.theriogenology.2010.02.016.

Silper, B. F., A. M. L. Madureira, L. B. Polsky, S. Soriano, A. F. Sica, J. L. M. Vasconcelos, and R. L. A. Cerri. 2017. Daily lying behavior of lactating Holstein cows during an estrus synchronization protocol and its associations with fertility. J. Dairy Sci. 100:8484–8495. https://doi.org/10.3168/jds.2016-12160.

Statistics Canada. 2020. Table 32–10–0130–01 Number of cattle, by class and farm type. Data table. https://doi.org/10.25318/3210013001-eng.

Vailes, L. D., and J. H. Britt. 1990. Influence of footing surface on mounting and other sexual behaviors of estrual Holstein cows. J. Anim. Sci. 68:2333–2339. https://doi.org/10.2527/1990.6882333x.

Wolpert, D. H. 1992. Stacked generalization. Neural Netw. 5:241–259. https://doi.org/10.1016/S0893-6080(05)80023-1.