Streamlining Quality Review of Mass Spectrometry Data in the Clinical Laboratory by Use of Machine Learning

Min Yu, MD, PhD; Lindsay A. L. Bazydlo, PhD; David E. Bruns, MD; James H. Harrison Jr, MD, PhD

Context.—Turnaround time and productivity of clinical mass spectrometric (MS) testing are hampered by time-consuming manual review of the analytical quality of MS data before release of patient results.

Objective.—To determine whether a classification model created by using standard machine learning algorithms can verify analytically acceptable MS results and thereby reduce manual review requirements.

Design.—We obtained retrospective data from gas chromatography–MS analyses of 11-nor-9-carboxy-delta-9-tetrahydrocannabinol (THC-COOH) in 1267 urine samples. The data for each sample had been labeled previously as either analytically unacceptable or acceptable by manual review. The dataset was randomly split into training and test sets (848 and 419 samples, respectively), maintaining equal proportions of acceptable (90%) and unacceptable (10%) results in each set. We used stratified 10-fold cross-validation in assessing the abilities of 6 supervised machine learning algorithms to distinguish unacceptable from acceptable assay results in the training dataset. The classifier with the highest recall was used to build a final model, and its performance was evaluated against the test dataset.

Results.—In comparison testing of the 6 classifiers, a model based on the Support Vector Machines algorithm yielded the highest recall and acceptable precision. After optimization, this model correctly identified all unacceptable results in the test dataset (100% recall) with a precision of 81%.

Conclusions.—Automated data review identified all analytically unacceptable assays in the test dataset, while reducing the manual review requirement by about 87%. This automation strategy can focus manual review only on assays likely to be problematic, allowing improved throughput and turnaround time without reducing quality.

Arch Pathol Lab Med. 2019;143:990–998; doi: 10.5858/arpa.2018-0238-OA

The use of mass spectrometric (MS) assays in clinical laboratories is increasing because of their advantages in analytical sensitivity and specificity, but sample handling and results reporting are commonly based on manual processes. Automation of sample preparation and data processing can improve productivity and efficiency, but manual review of gas chromatography–mass spectrometry (GC-MS) and liquid chromatography tandem MS data for each sample remains a time-consuming process because of the complexity of the output. Initial verification of runs requires evaluation of multiple features for each assay, including but not limited to, peak shape, retention time, several ion ratios, and area of internal standard (IS). After initial review, analyses of most samples are found to have acceptable characteristics and are released, while problematic results are reviewed in greater detail. A method to automate the screening of initial results and focus attention on runs that are most likely to need detailed review could improve turnaround times and laboratory efficiency.

Evaluation of the quality of MS analyses involves classification of assay results, based on relationships between the absolute and relative values of multiple features, such as those noted above. These values and their relationships are typically assessed in aggregate by experts in the laboratory who determine whether assays are acceptable for release. Criteria for acceptability are generally not expressed as specific interpretation guidelines or cutoff values for each metric, making it difficult to use simple rule-based automation for assay verification. Machine learning strategies are useful for automating classification problems dependent on relationships between multiple inputs and are widely used in many domains, including health care. Machine learning has been used previously with MS as an aid to interpretation in protein identification and biomarker discovery, where feature patterns in mass spectra were associated with particular protein structures.

While interest in machine learning in health care often focuses on its potential for aiding clinical decisions and predicting prognosis, machine learning also has excellent
potential for supporting complex decision-making related to health care workflow. In this study, we test whether a supervised machine learning strategy can be effective and efficient for classifying MS results as acceptable for release or requiring manual review. For training and test cases we used a library of clinical assays for 11-nor-9-carboxy-delta-9-tetrahydrocannabinol (THC-COOH) in urine that were previously classified by laboratory experts as acceptable or unacceptable for release. Supervised machine learning algorithms differ in their performance depending on the characteristics of the data used to train and test them, particularly when the number of cases is relatively low (about 1200 in our library). To account for these potential differences we evaluated the performance of 6 commonly used algorithms in creating a classification model and optimized the best-performing model to assess its suitability for use in a clinical MS workflow.

MATERIALS AND METHODS

GC-MS Analysis of THC-COOH in Urine Samples

The data for this study were derived from routine clinical samples previously analyzed for THC-COOH. Briefly, urine samples were subjected to base hydrolysis, extracted using a Bond-Elut CERTIFY solid phase extraction column (Agilent Technologies, Santa Clara, California), dried under nitrogen, and derivatized. Analysis was performed on 2 Agilent instruments (6890-5975 and 6890-5977) using the associated MassHunter software (Agilent Technologies). This assay has an analytical measurement range of 15 to 800 ng/mL. The ions used for THC-COOH identification included 371 for quantification, with 473 and 488 used as qualifier ions. The THC-COOH deuterated IS ions were 374 for quantification, with 476 and 491 used for qualification. Ion ratios were calculated from the following formula: Peak Area (Quantifier)/Peak Area (Qualifier), using the qualifying ion 473 for ion ratio 1 and qualifying ion 488 for ion ratio 2. The same approach was used to calculate the IS ion ratio, where 476 was used as qualifying ion 1 and 491 was qualifying ion 2.

Manual Result Verification

Results for each specimen were labeled as acceptable or unacceptable for release by manual review during routine laboratory operation. The review was carried out cooperatively by the performing technologist and the laboratory director. Review included consideration of the THC concentration result, retention time, IS peak area, relative ion ratios at 473 and 488, and relative ion ratios of the IS at 476 and 491. The tracing was reviewed with assessment of peak shape and relative retention time by eye. Ion ratios were flagged for attention if they differed by more than ±20% from the ion ratio of the midlevel calibrator. However, these flags did not individually determine the acceptability of a result, which was judged from multiple parameters. Other than these flagging guidelines, there were no specific cutoff values or rules that defined an acceptable versus unacceptable result. Unacceptable results were triaged for further review and possible intervention (such as repeated analysis) according to standard laboratory operating procedures.

Data Collection and Preparation

An overview of the machine learning workflow is shown in Figure 1. The initial dataset included all routine MS analyses for our in-house THC-COOH assay from September 2015 to December 2016, excluding results for quality control samples, calibrators, and patient samples with THC-COOH below the analytical measurement range. Metrics for each sample (Table 1) were calculated on preserved data by using the MassHunter software and included...
ion ratios, IS area, and relative retention time, which normalized the retention time of THC-COOH to that of the IS. Peak width, symmetry, and height to area ratio (THC-COOH) were calculated as metrics to substitute for manually judged peak shape. Patient identities and other identifying information such as service dates were excluded from the dataset. To minimize run-to-run variations, all ion ratios and IS area were normalized to the corresponding values produced by a midlevel calibrator (100 ng/mL) in the calibration curve. Thus, for each result, 10 quality-related characteristics (or “features” for machine learning, see below) of the GC-MS output were captured (Table 1). The set of features for each result was labeled as either nonoutlier or outlier on the basis of the original manual result verification (acceptable or unacceptable analytical results, respectively).

The full dataset was divided into training and test subsets by random partitioning at a fixed ratio, with two-thirds assigned to the training set and one-third assigned to the test set. The ratio of outlier to nonoutlier results was kept constant in the 2 sets. The training set was used for classifier selection and model training; the test set was withheld for later evaluation of the final model (Figure 1). Data processing, analytics, and machine learning were carried out with the Python programming language (Python version 3.6.5) with the included libraries Pandas version 0.23.0 for data processing and Scikit-learn version 0.19.1 for feature selection and machine learning (all analytics software was part of the Anaconda Python open source distribution from Anaconda, Inc., www.anaconda.com). The analytics software was installed on a Hewlett-Packard Model DC8300 desktop computer with 4-GB RAM running Windows 7 (64 bit, Microsoft Inc, Redmond, Washington).

### Preprocessing and Feature Selection

Some of the metrics of interest showed skewed distributions and therefore the data were standardized by preprocessing using the RobustScaler utility class from Scikit-learn with a specified quantile range of 20 to 80. This strategy is appropriate for data that are not normally distributed and that contain outliers. The method normalizes each metric to the value of the specified quantile range in the distribution of that metric and centers the new distribution on zero.10 The standardized features in the training data were ranked in relative importance by using 2 feature selection methods: a “wrapper” method based on a linear Support Vector Machines (SVM) algorithm with recursive feature elimination, and an “embedded” method using a random forests algorithm.23 This ranking provided a basis for pruning less important features during model development (Figure 1).

### Algorithm Selection and Model Evaluation

Classifiers were constructed by using 6 common supervised machine learning methods selected because they provided a broad range of machine learning strategies: logistic regression, SVM, decision tree, random forest, K-nearest neighbors, and ada-boost (Table 2). To compensate for unbalanced class representation, class weight was adjusted 9:1 to increase the penalty of misclassification to the minority (outlier) class. Classifier performance and parameter selection were assessed by using stratified 10-fold cross-validation11 with maintenance of target class percentages across the sets to limit fluctuation in minority class representation during cross-validation. Because there was a low incidence of outliers (~10%) and our goal was to identify all outliers at the expense of including some nonoutliers, we scored performance by recall of the outlier class (ie, sensitivity for detecting samples with results that had not been accepted during manual review; see Table 3) rather than the accuracy of assignment of both classes. Selection of the optimal classifier and additional manual tuning of its hyperparameters (analyst-definable, nonlearned variables) were based on the recall score in stratified 10-fold cross-validation. The impact of the feature set on classifier performance was assessed by comparing classifiers constructed with all 10 features against those using feature sets in which less important features were progressively pruned (6, 4, 3, and 2 features). The final classification model was built by training the selected machine learning algorithm with

### Table 1. Features of Mass Spectrometric Results

| Names                        | Description                                                                 |
|------------------------------|-----------------------------------------------------------------------------|
| Relative ion ratio 1 (ion 473) | Sample THC-COOH qualifier (473) ion ratio/calibrator 3 THC qualifier (473) ion ratio |
| Relative ion ratio 2 (ion 488) | Sample THC-COOH qualifier (488) ion ratio/calibrator 3 THC qualifier (488) ion ratio |
| Relative ion ratio 1 (IS) (ion 476) | Sample IS qualifier (476) ion ratio/calibrator 3 IS qualifier (476) ion ratio |
| THC concentration            | Calculated THC-COOH concentration                                            |
| Relative ion ratio 2 (IS) (ion 491) | Sample IS qualifier (491) ion ratio/calibrator 3 IS qualifier (491) ion ratio |
| Relative IS area a            | Sample IS peak area/calibrator IS peak area                                  |
| Peak symmetry a               | Sample THC-COOH peak symmetry                                               |
| Peak width b                  | Sample THC-COOH peak width                                                 |
| Peak height/area c            | Sample THC-COOH peak height/sample THC peak area                            |
| Relative retention time       | Sample THC-COOH retention time/sample IS retention time                     |

Abbreviations: IS, internal standard; THC, tetrahydrocannabinol; THC-COOH, 11-nor-9-carboxy-delta-9-tetrahydrocannabinol.

- a Peak integration used the default software integrator, which sets the base at 5% height.
- b Peak Symmetry = Lengthleaf/Lengthtotal, where the lengths are the horizontal distance from the peak apex to the front and tail of the peak at 10% height.
- c Peak width is measured at 50% of peak height.

### Table 2. Machine Learning Algorithms

| Algorithm          | Description                                                                 |
|--------------------|-----------------------------------------------------------------------------|
| Logistic regression| Regularized logistic regression based on liblinear,26 with a logistic link function |
| SVM                | SVM based on libsvm,27 with a radial basis function                         |
| Decision trees     | A CART algorithm with splits based on Gini impurity                        |
| Random forest      | Random forest of decision trees based on Breiman28 with 10 estimators and splits based on Gini impurity |
| K-Nearest neighbors| An instance-based classifier with class assignment from voting of the 5 nearest vectors, based on Euclidean distance |
| Ada-boosting       | An ensemble learning method based on AdaBoost-SAMMEm,25 in which a primary decision tree is supplemented by multiple additional trees that are iteratively improved by reweighting incorrectly classified examples |

Abbreviations: CART, classification and regression tree; SAMME, Stagewise Additive Modeling using a Multi-class Exponential Loss function; SVM, Support Vector Machines.

- a Data derived from scikit-learn documentation.25
RESULTS
Dataset
Our dataset included a total of 1267 deidentified patient results (Figure 1) with a nonoutlier to outlier label ratio of about 9:1, that is, about 90% of GC-MS results were judged acceptable for release and about 10% of results were judged unacceptable during previous manual review. The data were split by random partitioning at fixed ratio, with two-thirds assigned to a training dataset and the remaining one-third assigned to a test dataset. The label ratio of outliers to nonoutliers in the original data was maintained across the resulting training dataset of 848 results and test dataset of 419 results.

Feature Selection
A total of 10 quality-related features were available for each GC-MS result (Table 1). To reduce redundant information, reduce the likelihood of overfitting, and make machine learning simpler and faster, features are typically ranked in importance and less important features are “pruned” (removed from testing and training data). Feature importance is commonly ranked in preliminary studies of the impact of a feature’s absence on the accuracy of a simple classifier (eg, using a “leave-one-out” approach). To avoid the possibility of favoring a particular machine learning algorithm, based on the features selected, we evaluated the importance of features for the classification task by using 2 different simple classifiers.7,11 Table 4 lists the ranking of features by their impact on the accuracy of simple SVM and Random Forest classifiers. The top 4 features, namely, relative ion ratio 1 (ion 473), relative ion ratio 2 (ion 488), relative IS ion ratio 1 (ion 476), and calculated THC-COOH concentration, were identical for both feature selection methods (Table 4). This consensus of the 2 distinct feature selection methods suggests that these 4 features are generally important independently of the machine learning algorithm.

Algorithm Selection and Tuning
We built and tested preliminary models by using 6 different machine learning algorithms with the training dataset. Performance ranking was based primarily on the ability of the algorithm to identify results that required review (“recall” or sensitivity) and secondarily on ability to avoid erroneously identifying acceptable results as results that needed manual review (“precision” or positive predictive value). Recall metrics for each classifier, based on 10-fold cross-validation using the training dataset, are shown in Table 5. The performance of each classifier was tested with all 10 features (Table 4), and then with progressively restricted feature sets in which less important features were omitted (Table 5). Several of the algorithms showed best performance with 4 features, consistent with their previous identification as the consensus most important. All algorithms except logistic regression showed lower recall performance when restricted to only the 2 most important features (data not shown). SVM was robust to feature set restriction and showed the highest recall with all feature sets (Table 5); therefore, it was selected as the algorithm to carry forward for manual hyperparameter tuning (see Materials and Methods) in a second round of 10-fold cross-validation using the training data.

Final Model Performance
Since the intended use of the classifier was as a screening method that should identify all outlier runs at the expense of including some nonoutlier runs, tuning optimized recall over precision (see Materials and Methods). The tuned SVM model was re-trained on the full training dataset to yield the final model. This model was evaluated by using the held-out test dataset (Figure 1), and performance metrics were calculated for feature sets containing 10, 6, 4, and 3 features (Table 6). Metrics were also calculated for a 2-feature set, but performance was lower than for the larger feature sets (data not shown). Performance for the feature sets containing 6, 4, or 3 features was very close, with the 4-feature set showing optimal recall (100%) and a precision of 81%. Table 7 presents the confusion matrix for the 4-feature classifier, which shows no false negatives (100% recall) and 10 false positives (81% precision) in the test dataset of 419 results.

Characteristics of Features in Nonoutlier, Outlier, and False-Positive Assays
Distributions of the values of the 4 most important features in nonoutlier, outlier, and false-positive cases are shown in Figure 2. Nonoutlier (acceptable) assays had relatively tight distributions of values for all 4 metrics (Figure 2, top row) that were clearly distinct from outlier and false-positive cases. The distributions for the features were tight for both nonoutliers and outliers; therefore, these features were not selected for the classifier. The 4 features, namely, relative ion ratio 1 (ion 473), relative ion ratio 2 (ion 488), relative IS ion ratio 1 (ion 476), and calculated THC-COOH concentration, were identical for both feature selection methods (Table 4). This consensus of the 2 distinct feature selection methods suggests that these 4 features are generally important independently of the machine learning algorithm.

Algorithm Selection and Tuning
We built and tested preliminary models by using 6 different machine learning algorithms with the training dataset. Performance ranking was based primarily on the ability of the algorithm to identify results that required review (“recall” or sensitivity) and secondarily on ability to avoid erroneously identifying acceptable results as results that needed manual review (“precision” or positive predictive value). Recall metrics for each classifier, based on 10-fold cross-validation using the training dataset, are shown in Table 5. The performance of each classifier was tested with all 10 features (Table 4), and then with progressively restricted feature sets in which less important features were omitted (Table 5). Several of the algorithms showed best performance with 4 features, consistent with their previous identification as the consensus most important. All algorithms except logistic regression showed lower recall performance when restricted to only the 2 most important features (data not shown). SVM was robust to feature set restriction and showed the highest recall with all feature sets (Table 5); therefore, it was selected as the algorithm to carry forward for manual hyperparameter tuning (see Materials and Methods) in a second round of 10-fold cross-validation using the training data.

Final Model Performance
Since the intended use of the classifier was as a screening method that should identify all outlier runs at the expense of including some nonoutlier runs, tuning optimized recall over precision (see Materials and Methods). The tuned SVM model was re-trained on the full training dataset to yield the final model. This model was evaluated by using the held-out test dataset (Figure 1), and performance metrics were calculated for feature sets containing 10, 6, 4, and 3 features (Table 6). Metrics were also calculated for a 2-feature set, but performance was lower than for the larger feature sets (data not shown). Performance for the feature sets containing 6, 4, or 3 features was very close, with the 4-feature set showing optimal recall (100%) and a precision of 81%. Table 7 presents the confusion matrix for the 4-feature classifier, which shows no false negatives (100% recall) and 10 false positives (81% precision) in the test dataset of 419 results.

Characteristics of Features in Nonoutlier, Outlier, and False-Positive Assays
Distributions of the values of the 4 most important features in nonoutlier, outlier, and false-positive cases are shown in Figure 2. Nonoutlier (acceptable) assays had relatively tight distributions of values for all 4 metrics (Figure 2, top row) that were clearly distinct from outlier and false-positive cases. The distributions for the features were tight for both nonoutliers and outliers; therefore, these features were not selected for the classifier. The 4 features, namely, relative ion ratio 1 (ion 473), relative ion ratio 2 (ion 488), relative IS ion ratio 1 (ion 476), and calculated THC-COOH concentration, were identical for both feature selection methods (Table 4). This consensus of the 2 distinct feature selection methods suggests that these 4 features are generally important independently of the machine learning algorithm.

Algorithm Selection and Tuning
We built and tested preliminary models by using 6 different machine learning algorithms with the training dataset. Performance ranking was based primarily on the ability of the algorithm to identify results that required review (“recall” or sensitivity) and secondarily on ability to avoid erroneously identifying acceptable results as results that needed manual review (“precision” or positive predictive value). Recall metrics for each classifier, based on 10-fold cross-validation using the training dataset, are shown in Table 5. The performance of each classifier was tested with all 10 features (Table 4), and then with progressively restricted feature sets in which less important features were omitted (Table 5). Several of the algorithms showed best performance with 4 features, consistent with their previous identification as the consensus most important. All algorithms except logistic regression showed lower recall performance when restricted to only the 2 most important features (data not shown). SVM was robust to feature set restriction and showed the highest recall with all feature sets (Table 5); therefore, it was selected as the algorithm to carry forward for manual hyperparameter tuning (see Materials and Methods) in a second round of 10-fold cross-validation using the training data.

Final Model Performance
Since the intended use of the classifier was as a screening method that should identify all outlier runs at the expense of including some nonoutlier runs, tuning optimized recall over precision (see Materials and Methods). The tuned SVM model was re-trained on the full training dataset to yield the final model. This model was evaluated by using the held-out test dataset (Figure 1), and performance metrics were calculated for feature sets containing 10, 6, 4, and 3 features (Table 6). Metrics were also calculated for a 2-feature set, but performance was lower than for the larger feature sets (data not shown). Performance for the feature sets containing 6, 4, or 3 features was very close, with the 4-feature set showing optimal recall (100%) and a precision of 81%. Table 7 presents the confusion matrix for the 4-feature classifier, which shows no false negatives (100% recall) and 10 false positives (81% precision) in the test dataset of 419 results.

Characteristics of Features in Nonoutlier, Outlier, and False-Positive Assays
Distributions of the values of the 4 most important features in nonoutlier, outlier, and false-positive cases are shown in Figure 2. Nonoutlier (acceptable) assays had relatively tight distributions of values for all 4 metrics (Figure 2, top row) that were clearly distinct from outlier and false-positive cases. The distributions for the features were tight for both nonoutliers and outliers; therefore, these features were not selected for the classifier. The 4 features, namely, relative ion ratio 1 (ion 473), relative ion ratio 2 (ion 488), relative IS ion ratio 1 (ion 476), and calculated THC-COOH concentration, were identical for both feature selection methods (Table 4). This consensus of the 2 distinct feature selection methods suggests that these 4 features are generally important independently of the machine learning algorithm.
positive distributions. Metrics from outlier (unacceptable) assays had broader distributions that overlapped the distributions of nonoutlier assays, but also had prominent subsets of elevated and depressed values (Figure 2, middle row). False positives also showed substantial overlap with nonoutlier runs, but with broader distributions that contained some distinct subsets that resembled parts of the outlier assay distributions (Figure 2, bottom row).

Plots of the values of the 4 metrics in individual cases show that they vary relatively independently in outlier assays and false positives (Figure 3, A and B, respectively). A variety of different patterns are present across the range of outlier cases, including individual extreme values as well as cases with more modest deviations in all 4 metrics (Figure 3, A). False-positive cases did not show a particular characteristic pattern (Figure 3, B), though the deviations overall were relatively modest, compared to the outlier cases.

SVM models define a multidimensional barrier called a "hyperplane" between distinguishable clusters of cases in the training set, and new cases are classified by their position relative to the hyperplane. The hyperplane and clusters can be visualized for models incorporating up to 3 features (ie, 3 dimensions). Higher-dimensional models operate in an analogous way but cannot be easily visualized. A visualization of the cases and hyperplane for the 3-feature SVM model (Table 6) is shown in Figure 4. The mesh represents the hyperplane with the nonoutlier assays visible as a cluster of filled points inside the hyperplane, and the outlier assays are shown as open circles external to the hyperplane. The outlier assays do not appear to form a distinct cluster in our data, though diffuse clusters might appear if more data were available. False positives are shown as filled points that are outside the hyperplane. The false-positive points also do not form a distinct cluster. These results suggest a relatively diffuse spread of outlier cases that can vary in multiple ways around a tight cluster of relatively uniform acceptable cases.

### DISCUSSION

The goal in this work was to develop a strategy for automating the classification of GC-MS results for THC-COOH as acceptable for reporting or as unacceptable and requiring manual review. We wished to develop a system that performed similarly to the current manual review of all assays, and we used data previously labeled by manual review for training and testing our models. We were able to duplicate the performance of the manual review reasonably closely, with 100% recall of unacceptable results in our 4-feature model (Table 6). If applied in a production setting, this strategy could substantially reduce the number of results needing manual review and improve workflow and turnaround time by directing attention to a limited number of results needing detailed evaluation.

To our knowledge, the use of machine learning algorithms for verification of GC-MS results has not been published previously. Machine learning, including some of the algorithms we evaluated, has been used previously for MS biomarker identification.5–9,12–18 More generally in the clinical laboratory, machine learning has been used recently.

| Table 5. Effect of Feature Pruning on the Recall of Machine Learning Algorithms for Outlier Cases in the Training Dataseta |
|------------------|-------|------------------|------------------|------------------|
| Classifierb | Ten Featuresc Mean (SD)b | Six Featuresd Mean (SD) | Four Featurese Mean (SD) | Three Featuresf Mean (SD) |
| SVM | 0.92 (0.10) | 0.91 (0.10) | 0.90 (0.09) | 0.91 (0.10) |
| Logistic regression | 0.85 (0.11) | 0.86 (0.11) | 0.88 (0.08) | 0.74 (0.17) |
| Ada boosting | 0.81 (0.10) | 0.78 (0.11) | 0.80 (0.12) | 0.76 (0.05) |
| K-Nearest neighbors | 0.76 (0.10) | 0.84 (0.12) | 0.85 (0.12) | 0.81 (0.13) |
| Decision tree | 0.74 (0.12) | 0.78 (0.18) | 0.77 (0.20) | 0.77 (0.21) |
| Random forest | 0.65 (0.17) | 0.72 (0.16) | 0.76 (0.22) | 0.77 (0.18) |

Abbreviations: IS, internal standard; SD, standard deviation; SVM, Support Vector Machines.

a n = 848 results.
b Ordered by recall performance, using all 10 features.
c All 10 features used (Table 4).
d Includes the first 4 features from Table 4, plus the fifth feature identified by each of the 2 ranking methods (relative retention time and relative ion ratio 2 [IS]).
e Includes the first 4 features from Table 4.
f Includes the first 3 features from Table 4.

| Table 6. Effect of Feature Pruning on the Performance of the Optimized SVM Model on the Test Datas |
|------------------|-------|------------------|------------------|------------------|
| No. of Features | Recall | Precision | F1 Score |
| Tenb | 0.98 | 0.77 | 0.86 |
| Sixc | 0.98 | 0.83 | 0.90 |
| Fourd | 1.00 | 0.81 | 0.90 |
| Threee | 0.98 | 0.81 | 0.89 |

Abbreviations: IS, internal standard; SVM, Support Vector Machine.
a n = 419 results, distinct from the training set.
b All 10 features used (Table 4).
c Includes the first 4 features from Table 4, plus the fifth feature identified by each of the 2 ranking methods (relative retention time and relative ion ratio 2 [IS]).
d Includes the first 4 features from Table 4.
e Includes the first 3 features from Table 4.

| Table 7. Confusion Matrix for the Optimized SVM Model on the Test Datas |
|------------------|-------|------------------|------------------|
| | Outlier | Nonoutlier |
| Outlier (predicted) | 44 (TP) | 10 (FP) |
| Nonoutlier (predicted) | 0 (FN) | 365 (TN) |

Abbreviations: FN, false negative; FP, false positive; SVM, Support Vector Machine; TN, true negative; TP, true positive.
a n = 419 results, distinct from the training set.
to assess the likelihood of particular clinical contexts associated with patterns in laboratory test results and to identify dependencies between test results.²⁹,³⁰ Automated review of MS quality data has been supported previously in a limited number of nonlearning, rules-based systems.³¹ In these systems users individually define acceptability criteria for numerous metrics such as retention time, ion ratios, and IS responses. Managing these multiple rules and cutoffs can be challenging, particularly when dependencies between the metrics exist. By contrast, machine learning algorithms can integrate the evaluation of multiple attributes of the data (including relationships among data elements) and link these patterns to quality judgments that are locally relevant. Given appropriately classified historical data such as ours, it is relatively easy using machine learning techniques to generate and validate an assay-specific verification model based on multiple quality-related features. Model creation and validation using modern processing pipelines (the steps in Figure 1 beginning with Data Partitioning) may require less effort than development and validation of a rules-based system incorporating a similar number of metrics.

Our dataset included 10 features commonly considered to be associated with the quality of a clinical MS result (Table 1). The features are interdependent, and specific cutoffs per feature have not been established that would allow a set of simple rules to define an acceptable result. Under routine conditions in our laboratory the features for each run are reviewed subjectively by the technician and laboratory director to determine run acceptability. This set of 10 features provided a starting point for a machine learning feature set. It is good practice in machine learning to evaluate the importance of features before model training and eliminate unimportant features (“feature selection”). Feature selection methods rank features according to their ability to influence the performance of a basic classifier (ie, their importance for correct classification). Features may be ranked low because they are not helpful in making a classification decision or because their information is redundant with other features. The benefits of feature selection include a reduction in training and processing time without a reduction in model accuracy, and a decrease in the likelihood of “overtraining,” that is, including irrelevant patterns in unimportant features that may reduce model accuracy.³² Thus a goal in machine learning model development is to use the smallest feature list consistent with good classifier performance.³³ Note that while feature selection identifies the features most likely to contribute to correct model output, the ranking does not necessarily reflect the relative importance or weights of those features in the final trained model. We used 2 feature selection methods based on different simple classifiers to rank the importance of the 10 features (Table 4). This ranking was
We tested 6 commonly used algorithms with distinct performance across datasets that differ in size, number of features, ratio of the number of features to the total number of items, and variability of features. Hence it is reasonable to evaluate the performance of several algorithms to determine which is optimal for a particular dataset and use. We tested 6 commonly used algorithms with distinct characteristics for their ability to classify GC-MS assay quality. Our initial evaluation focused on recall for unacceptable assays, using the complete 10-element feature list compared with pruned feature lists containing 6, 4, 3, and 2 features. The algorithms yielded a range of performance; SVM was chosen for further development because it produced the highest recall for all feature lists and its performance was robust to feature pruning (Table 5). After optimization by hyperparameter tuning, SVM trained on 4 features correctly identified all samples in the test dataset that required manual review (recall 100%; Tables 6 and 7) while falsely labeling only 2.7% of acceptable results as needing review (Table 7).

This classification task entailed identifying a minority (≈10%) of cases as belonging to the outlier class. In imbalanced datasets, machine learning algorithms are designed to maximize correct classification of the majority class. By contrast, our goal was to develop a screening tool that correctly classifies all outliers (high recall), and an additional limited number of false positives is acceptable.

When the machine learning goal is accurate identification of the minority class, it is important to (1) stratify datasets (eg, training data including cross-validation subsets, and test data) such that the proportion of the minority class is equal and representative, (2) weight the minority class during learning to compensate for its lower prevalence, and (3) use performance metrics appropriate for the goal, for example, recall for the minority class rather than overall accuracy. We incorporated all 3 of these approaches. The recall and precision of this model (100% and 81%, respectively) indicate that in a collection of 100 runs with 10 true outlier results we would reduce the manual review requirements from 100 results to about 13 results. Within the 13 results identified as outliers requiring review, we would find all 10 true outliers and 3 results that were not true outliers. This level of performance meets our original goals of comparability to routine human screening and significant reduction in review effort.

There was a clear distinction between the distributions of values of the features in nonoutlier versus outlier assays (Figure 2). Acceptable (nonoutlier) assays showed relatively tight distributions of all features, while unacceptable (outlier) assays showed broader distributions with prominent subpopulations at more extreme values. It is possible that these subpopulations may be clues to particular technical problems. The largest subpopulation for each feature in the unacceptable assays overlapped the tight distributions of the acceptable runs, suggesting that rejection of assays often was based on aberrant values for a limited number of features, with other features frequently similar to acceptable results. Unacceptable cases showed a variety of patterns in feature values (Figure 3, A), suggesting that the features may vary relatively independently. A plot of the feature space for the 3-feature classifier (Figure 4) does not show clear clustering of unacceptable cases that would suggest characteristic feature patterns. These results are consistent with our feature selection strategy in which features covarying with an important feature are down-ranked and pruned to eliminate redundancy. Although extreme values of features were frequent in unacceptable cases, some cases had moderate values of all features (Figure 3, A; and cases near the hyperplane in Figure 4), indicating that, in addition to extreme values, unacceptable cases may be characterized by patterns of borderline values in multiple features. Evaluation of these types of patterns in greater detail, and evaluation of a possible association of subpopulations of feature values with defined technical problems, will require a greater volume of data with prospective analysis and identification of technical problems as they occur.

The distributions of feature values in “false positives” (assays called acceptable in manual evaluation but identified as unacceptable by the classifier; Table 7) resembled the unacceptable assay feature distributions, though not all subpopulations were apparent (Figure 2). Inspection of the individual cases (Figure 3, B) reveals values that are less extreme with a variety of feature patterns. There is no apparent characteristic pattern associated with false positives, and the feature space plot for the 3-feature model did not show clustering of false positives suggestive of a common pattern (Figure 4). It is reasonable to speculate that manual review of borderline cases without specific guidelines may not distinctly separate acceptable from unacceptable cases. Our machine learning strategy is optimized for recall of unacceptable cases (see Materials and Methods). Dotted line: relative ion ratio 1; dashed line: relative ion ratio 2; dash-dotted line: relative ion ratio 1 internal standard; solid line: tetrahydrocannabinol concentration.

Figure 3. Feature patterns in outlier cases (A) and false positives (B) in the test dataset. Values are standardized to the interquartile interval for quantiles 20 to 80 of each feature and centered on 0 (see Materials and Methods). Dotted line: relative ion ratio 1; dashed line: relative ion ratio 2; dash-dotted line: relative ion ratio 1 internal standard; solid line: tetrahydrocannabinol concentration.
Nonoutlier and outlier class separation by SVM. The graph displays the SVM model trained on the 3 most important features (Table 4). The mesh represents the optimal "hyperplane" for separating the classes, as determined by the trained model. Cases within the hyperplane are classified as nonoutliers by the model and cases outside it are classified as outliers. Filled points represent cases manually defined as nonoutliers and open points were manually defined as outliers. Filled points outside the hyperplane are false positives. Values are standardized to the interquantile interval for quantiles 20 to 80 of each feature (see Materials and Methods). Abbreviations: IS, internal standard; SVM, Support Vector Machines.
validation of the machine learning model, using preclassified local datasets of limited size. Formal best practices and requirements for validation and performance monitoring of machine learning software do not yet exist. Once they have been established, it may be possible for sites to share classifiers such as those developed here after data compatibility is demonstrated and model performance is verified.

This initial work with a THC-COOH assay provides a proof of principle for application of machine learning to automated review of clinical MS data. Our study shows that machine learning using an SVM model for quality review of MS data can have excellent recall and good precision for distinguishing data that need review from data that can be released without review. Given the range of machine learning algorithms and techniques available we believe it is likely that this general approach is broadly applicable, and that automatic verification systems based on machine learning can be developed for a range of clinical MS assays that will improve laboratory productivity and workflow while maintaining or improving the quality of reported results.

References

1. Jannetto PJ, Fitzgerald RL. Effective use of mass spectrometry in the clinical laboratory. Clin Chem. 2016;62(1):92–98.
2. Leung KS, Feng BM. LC-MS/MS in the routine clinical laboratory: is its time come? Anal Bioanal Chem. 2014;406(9-10):2289–2301.
3. Zhang YV, Rockwood A. Impact of automation on mass spectrometry. Clin Chem Acta. 2015;450(23 October):298–303.
4. Jordan MI, Mitchell TM. Machine learning: trends, perspectives, and prospects. Science. 2015;349(6245):255–260.
5. Swan AL, Mobasheri A, Allaway D, Liddell S, Bacardit J. Application of machine learning to proteomics data: classification and biomarker identification in postgenomics biology. OMICS. 2013;17(12):595–610.
6. Barla A, Jurman G, Riccadonna S, Meiler S, Chierici M, Furlanello C. Machine learning methods for predictive proteomics. Brief Bioinform. 2008;9(2):119–128.
7. Datta S. Feature selection and machine learning with mass spectrometry data. Methods Mol Biol. 2013;1007:237–262.
8. Petrova NV, Wu CH. Prediction of catalytic residues using support vector machine with selected protein sequence and structural properties. BMC Bioinformatics. 2006;7:312.
9. Zhang L, Shao C, Zheng D, Gao Y. An integrated machine learning system to computationally screen protein databases for protein binding peptide ligands. Mol Cell Proteomics. 2006;5(7):1224–1232.
10. Scikit-learn. Preprocessing.RobustScaler—scikit-learn documentation. http://scikit-learn.org/stable/modules/generated/scikit-learn.preprocessing.RobustScaler.html. Accessed October 21, 2018.
11. Scikit-Learn. Model_selection.Stratifiedkfold—scikit-learn documentation. http://scikit-learn.org/stable/modules/generated/scikit-learn.model_selection.StratifiedKFold.html. Accessed October 21, 2018.
12. Bloemen K, Van Den Heuvel R, Govarts E, et al. A new approach to study exhaled proteins as potential biomarkers for asthma. Clin Exp Allergy. 2011;41(3):346–356.
13. Sudim BL, Qu Y, Davis JW, et al. Serum protein fingerprinting coupled with a pattern-matching algorithm distinguishes prostate cancer from benign prostate hyperplasia and healthy men. Cancer Res. 2002;62:3609–3614.
14. Izmirlian G. Application of the random forest classification algorithm to a SELDI-TOF proteomics study in the setting of a cancer prevention trial. Ann N Y Acad Sci. 2004;1020:154–174.
15. Purohit PV, Rocke DM. Discriminant models for high-throughput proteomics mass spectrometer data. Proteomics. 2003;3(9):1699–1703.
16. Saltan GA, Datta S, Moura H, Woolfitt AR, et al. Standardization and denoising algorithms for mass spectra to classify whole-organism bacterial specimens. Bioinformatics. 2004;20(17):3128–3136.
17. Wu B, Abbott T, Fishman D, et al. Comparison of statistical methods for classification of ovarian cancer using mass spectrometry data. Bioinformatics. 2003;19(13):1636–1643.
18. Zhu W, Wang X, Ma Y, Rao M, Glimm J, Kovach JS. Detection of cancer-specific markers amid massive mass spectral data. Proc Natl Acad Sci U S A. 2003;100(25):14666–14671.
19. Richardson A, Signor BM, Lidsbury BA, Badrick T. Clinical chemistry in higher dimensions: machine-learning and enhanced prediction from routine clinical chemistry data. Clin Biochem. 2016;49(16-17):1213–1220.
20. Luo Y, Szolovits P, Dighe AS, Baron JM. Using machine learning to predict laboratory test results. Am J Clin Pathol. 2016;145(6):778–786.
21. TBH Creative & Indigo Bioautomation. http://www.indigobio.com/ascent/. Accessed October 21, 2018.
22. Sheikhpour R, Saram MA, Gharaghani S, Chahooki MAZ. A survey on semi-supervised feature selection methods. Pattern Recognit. 2017;64(April):141–158.
23. Khondoker M, Dobson R, Skirrow C, Simmons A, Stahl D. A comparison of machine learning methods for classification using simulation with multiple real datasets from mental health studies. Stat Methods Med Res. 2016;25(5):1804–1823.
24. Rich C, Alexandru N-M. An empirical comparison of supervised learning algorithms. In: Proceedings of the 23rd International Conference on Machine Learning. New York: ACM; 2006:161–168.
25. Scikit-learn. http://scikit-learn.org/. Accessed October 21, 2018.
26. Wang X-R, Chang K-W, Hsieh C-J, et al. LIBLINEAR—a library for large linear classification. The Machine Learning and Data Mining Group at National Taiwan University. https://www.csie.ntu.edu.tw/~cjlin/liblinear/. Accessed October 21, 2018.
27. Chang C-C, Lin C-J. LIBSVM—a library for support vector machines. https://www.csie.ntu.edu.tw/~cjlin/libsvm/. Accessed October 21, 2018.
28. Breiman L. Random forests: machine learning. 2001;45(1):5–32.
29. Zhu J, Zou H, Rosset S, Hastie T. Multi-class AdaBoost: statistics and its interface. 2009;2(3):349–360.