Genetic profiling of olives for location of origin and variety discrimination using Machine Learning

Michael E. Mavroforakis
IEEE Senior Member
InTTrust S.A.
Athens, Greece
mmavrof@inttrust.gr

Harris V. Georgiou
Data Science Lab
University of Piraeus (UniPi)
Piraeus, Greece
hgeorgiou@unipi.gr

***Abstract*—Genetic profiling via biomarkers in the food industry is a technology that gains momentum in the context of quality assurance and protection against fraud, as well as securing commercial assets like designation of origin. However, current solutions are based on methods that require significant computational resources and management of large data volumes, making them unsuitable for applications in the context of Internet-of-Things (IoT), edge computing and microcontrollers (MCU). This study presents a novel, computationally efficient and robust approach for fully field-integrated, low-complexity and high-accuracy classification of olives variety and location of origin, based on genetic 'fingerprinting' via a minimal set of information-rich features. The method is tested with real-world datasets, achieving accuracy rates above 96% and 99%, respectively, using various instance-based and tree ensemble classification models.**

***Index Terms*—machine learning, genetic profiling, DNA analysis, food industry**

I. Introduction

Genetic profiling via data-driven methods is becoming more reliable and readily available for a wide range of industries, especially in the food supply chain. The genome itself is treated as the input source for generating DNA data for analysis via algorithms end methods from the Bioinformatics and Machine Learning (ML) domains. However, the volume and scale of the generated data are often a prohibitive factor for designing both efficient and simple models. On the other hand, the wide adoption of such processes in real-world production lines and supply chains requires economic and robust model designs. Moreover, in the context of massively field-deployed modules regarding Internet-of-Things (IoT), edge computing and microcontroller (MCU) applications, these solutions have to be of low-complexity, capable of running in hardware-constrained environments and even completely in offline mode (non-networked).

The purpose of this work is to formulate and present a completely data-driven and practical approach for genetic profiling of olives. This profiling is based on Polymerase Chain Reaction (PCR) for DNA duplication and High Resolution Melt (HRM) [1], [2] process for producing fluorescence-versus-temperature experimental data, in order to quantitatively characterize specific portions of the genome that are identified and highlighted by specially designed biomarkers. In practice, any similar genetic profiling is compatible with this data-driven approach, such as the DNA helicity-versus-temperature that has been used extensively in related works [2], [3]. However, the fluorescence modality was chosen as more reliable and easier to implement in the field, i.e., within the food supply chain itself and with low-end computing resources, compared to alternative methods already studied in the context of DNA helicity [3]–[5].

Olives (leaves) variety and location of origin were selected as the practical use case, since olive oil and particularly Extra Virgin Olive Oil (EVOO) is one of the most important agricultural products in Mediterranean countries and a high-risk target of non-compliances and frauds such as admixtures with other lower quality oils. There is already evidence of genetic diversity in olives genome at the scope of the Mediterranean region and beyond [6]. However, with this work it is the first time that the detection of the olives variety and (especially) the location of origin at such geographic granularity (distinct regions of Crete island, Greece) via purely data-driven genetic profiling is demonstrated at such high success rates.

The paper is organized as follows: section I provides a brief introduction to the context and the main challenges of the problem; section II presents the details of the problem, the exact goals of the proposed approach and the design limitations and constraints; section III describes the two main stages of the proposed approach, namely the re-processing pipeline and the classification models; section IV presents the experimental protocol and results; section V discusses the main outcomes and provides hints for future work; finally, section VI provides the main conclusions.

II. Problem Statement

The challenge addressed by the proposed method can be summarized as follows:

1) **Given:** A pool of data series of fluorescence versus temperature, produced by HRM-based analysis of DNA strands of olives (leaves),

This work was partially funded by S3FOOD Sub-Grant Agreement N VV106/2021, for the project 'DNA digitization: olive oil authenticity and traceability from field to bottle'. S3FOOD has received funding from the EU Horizon 2020 research and innovation programme under Grant Agreement No 824769.

2) **Produce:** A data-driven, computationally efficient processing sequence for 'fingerprinting' and discriminating these DNA strands with regard to: (a) their variety and (b) their location of origin.

The modality of fluorescence was based on the application of biomarkers specifically designed for this purpose, i.e., highlight the subtle genetic differences between different olives varieties and locations of origin in Crete, Greece, and with a data generation framework designed and implemented by InTTrust S.A. and BIOCOS P.C. [7]. Both these classification tasks are directly related to Protected Designation Origin (PDO) and Protected Geographical Indication (PGI) for quality assurance and protection against fraud in the food industry. This HRM-based data generation process is expected to be optimized, streamlined and widely available at the source of the food supply chain within the next few years [7].

There are three main requirements for the design of the proposed approach, which essentially constitute the technical limitations and constraints for the solution:

- Robust genetic profiling, that can be used as a 'fingerprint' of the source product throughout the food supply chain;
- Lightweight models, readily applicable to Internet-of-Things (IoT), edge computing and microcontroller (MCU) applications;
- Production-grade performance, i.e., highly efficient and accurate model proposals.

The first item is related to reliable identification and tracking of olives, as a use case, end-to-end in the food supply chain of related dietary products. In practice, the genetic 'fingerprint' of olive leaves should be able to distinguish important characteristics such as variety and location of origin, clearly unique and verifiable against admixtures and other impurities related to quality control and assurance. For a real-world application, the subtle genetic differences highlighted by the HRM-based DNA analysis must be translated into a well-defined data-centric processing sequence, which includes pre-processing and can support automated classification. Such applications may employ other cutting-edge technologies, such as blockchain and smart contracts [7], for provable and tamper-proof authenticity and purity of the food product.

The second item is a design choice that enables the implementation of such applications in hardware-constrained environments, specifically in relation to computing resources. If such an end-to-end process, implementing genetic profiling and 'fingerprinting' of food products, can be developed in low-cost, low-energy and even disposable computing modules, then it becomes more relevant to real-world massive deployments in the food supply chain of agriculture and food products. That is, instead of tracking only lot numbers and distribution batches, the product itself is characterized by intrinsic biometric-like data generators.

Finally, the third item is also a design choice in terms of being 'provably' efficient for real-world applications, via extensive experimental tests and model optimizations. This means that it is not enough to have relatively high peak success rates in the classifications, but require consistently high mean success rates along with robustness and generalization level. The two modalities presented in this work, i.e., variety and location of origin, have been selected precisely for this reason. Other similar classification targets and tasks are still under refinement for reaching the same output quality in terms of predictive accuracy.

III. Methods

The output of the HRM-based DNA analysis is data series of fluorescence values versus temperatures, spanning up to several thousands of data points each w.r.t. a range of temperatures ($T \in [70^oC \dots 95^oC]$, sampled per 0.01^oC). These data series are further processed via signal analysis to extract relevant features and to train predictive ML models, characterizing and identifying the content of each sample.

More specifically, this processing consists of two distinct stages: (i) a hybrid pre-processing pipeline for data processing and features generation and selection; and (ii) training and evaluation of predictive ML models for each classification task, as described in section IV.

A. Pre-processing pipeline

For the pre-processing stage, the term 'hybrid' refers to the complementary involvement of human expert and automated cleansing during the data quality review, especially regarding the assessment of extreme cases and error samples that are removed. This is the standard procedure for data quality improvement within a Data Analytics (DA) pipeline, involving human experts specifically in the design phase, in order to formulate clear and efficient validation rules and thresholds for the subsequent fully-automated system. In practice, the human expert labels the validity of each data sample, flagging any obvious errors for removal before any further processing, i.e., to avoid contaminating the generated feature data for training with invalid instances.

The complete pre-processing pipeline consists of the following steps:

1) Raw data series import
2) Visual inspection of the data, removal of errors
3) Data filtering for noise removal
4) 1st derivative data series generation
5) Data corrections / pre-fitting adjustments (prefix)
6) Curve-fitting functions (Gaussian, rational)
7) Curve-fitting quality metrics (RMSE)
8) Visual assessment / validation of curve-fitting
9) Statistical filtering for curve-fitting quality
10) Feature vectors to training dataset

Steps 1-4 constitute the import, inspection, noise removal via low-pass filtering (smoothing) and preparation of the data for further processing. Figure 1 is an example of the data series for fluorescence versus temperature using a specific biomarker; the two colors illustrate the discrimination between two varieties, namely 'Koroneiki' (*'vK'*) and 'Tsounati' (*'vT'*), in olive leaves.

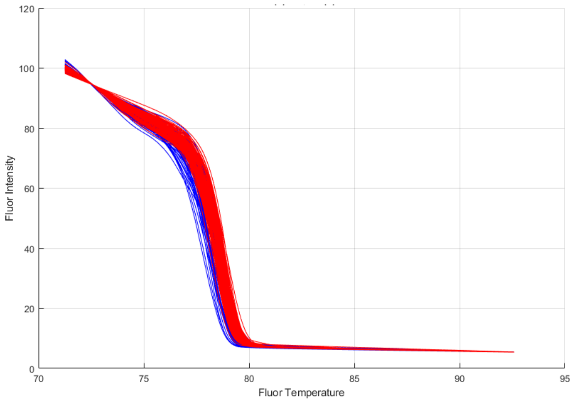

Fig. 1. Pre-processed data series of DNA fluorescence versus temperature in two varieties (red=*vK*, blue=*vT*) for olive leaves.

Step 5 is a common practice when working with 1st derivative data from temperature-based DNA profiling, in order to compensate for non-horizontal of temperature response below the critical threshold of phase transition, as illustrated in Figure 1 at about 78 degrees (C). In turn, this negative linear slope translates to a stable negative value in the 1st derivatives plot, illustrated in Figure 2.

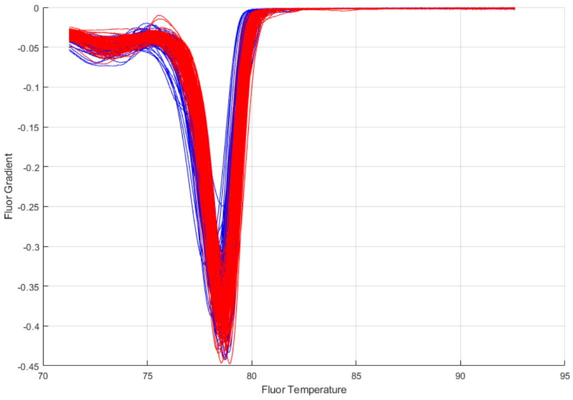

Fig. 2. Pre-processed 1st derivative series of DNA fluorescence versus temperature in two varieties (red=*vK*, blue=*vT*) for olive leaves.

Due to this inconsistency in the phase transitions of temperature response, the two regions of constant-value 1st derivative (i.e., on the left and the right of the middle region) are fixed at different levels, negative for the left region and usually near zero for the right region. This shape is often incompatible with symmetric curve-fitting models like the Gaussian. Hence, a simple k-order weighting prefix correction can be applied, according to (1), which also mitigates any remaining noise artifacts of larger scale, which are often observed in this left region of the 1st derivative curve:

$$\widehat{y'_t} = y'_t\,(t/\tau)^k\,,\ \forall\, 0 \le t \le \tau \tag{1}$$

In step 6, two curve-fitting models have been employed for encoding the most important properties of each 1st derivative data series. Specifically, a Gaussian function as in (2) and a rational polynomial function as in (3) have been selected, based on their efficiency, easy convergence and limited number of parameters.

$$f_g(x) = \alpha\, e^{\left(\frac{x-\mu}{\sigma}\right)^2} \tag{2}$$

$$f_r(x) = \frac{p_1 x + p_2}{x^2 + q_1 x + q_2} \tag{3}$$

Figure 3 illustrates an example of rational curve-fitting on a 1st derivative data series sample, with prefix weighting disabled (not necessary).

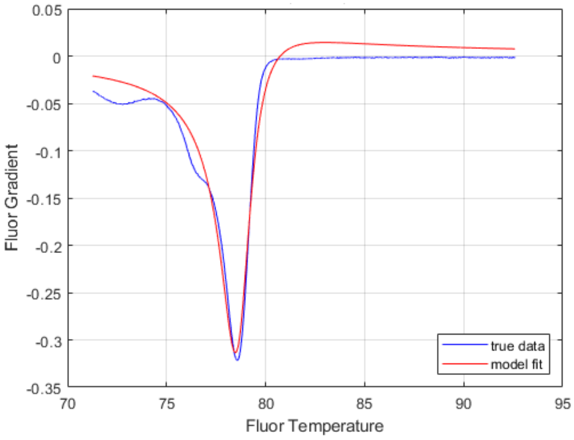

Fig. 3. Example of the 1st derivative data series and the corresponding curve fitting function (rational), with prefix (left) disabled.

Steps 7-9 constitute the quality assessment and filtering of the results produced by the curve-fitting step. More specifically, the Root Mean Squared Error (RMSE) was employed as a robust quality metric with bias towards larger deviations, i.e., more homogeneous errors are preferred over results with the same mean but with larger variance (e.g. a few large extremes). For each curve-fitting function f, the resulting $RMSE_f$ is kept as a separate feature, as well as used in combination (multiplied together) to produce a single quality index, namely $RMSE_{(g,r)} = RMSE_g \cdot RMSE_r$.

Next, the curve-fitted data series samples (1st derivative) is sorted against this $RMSE_{(g,r)}$ index, with visual assessment and validation identifying the first conclusive threshold that separates ‘good’ from ‘bad’ fits, i.e., the sudden transition between the sharp drop from large fitting errors to relatively low and stable fitting errors. This assessment and threshold estimation can be easily automated via linear slope analysis, but in this study it was purposely ‘hybrid’ with visual inspection in the loop, in order to rule out any effects that could poison the quality and reliability of the next stage, i.e., the construction of the training and testing datasets for the classification models.

The following plot is an example illustrating the curve-fitting quality assessment based on the combined error metric $RMSE_{(g,r)}$ as previously described. In this case, for the data series generated with a specific marker, the two curve-fitting functions produce acceptable fits for all but the first 17 (X-axis) data samples, which are subsequently removed with the proper threshold value (Y-axis).

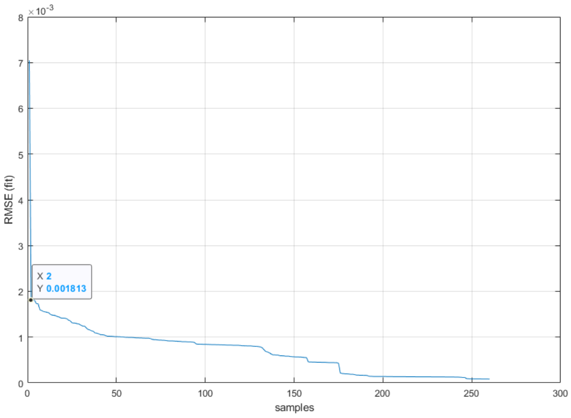

Fig. 4. Curve-fitting quality assessment based on the combined $RMSE_{(g,r)}$. In this example, the curve fitting functions produced acceptable fits for all but the first 17 (X-axis) data samples, which were discarded.

Finally, in step 10 the curve-fitting parameters for each data sample, as well as a few additional descriptive statistics, are grouped together into a concise feature vector that fully identifies each sample as a genetic 'fingerprint'.

Table I presents all the elements of this feature vector, which constitutes the initial dimensionality of the input space for the next stage, i.e., the design, training and testing of the classification models. The 'Data' column defines source data for evaluation ('X': external, 'S': Fluo-vs-Temp data series, 'D': 1st derivarive of S); 'FVxx' parameters correspond to the definitions in (2) and (3); and entropy according to (4):

$$f_{ent}(x) = -\sum_{i} p_x(i) \log p_x(i) \tag{4}$$

TABLE I
CONTENTS (DIMENSIONS) OF THE FEATURE VECTORS GENERATED BY THE PRE-PROCESSING PIPELINE.

Symbol	Data	Description
MK	X	Biomarker used $\{1,2,3\}$
MIN	S	Minimum value
MAX	S	Maximum value
ENT	S	Entropy (4)
FVg1	D	Gaussian fit, scale (α)
FVg2	D	Gaussian fit, mean (μ)
FVg3	D	Gaussian fit, sigma (σ)
FVg4	D	Gaussian fit, error ($RMSE_g$)
FVq1	D	Rational poly fit, numerator 1 (p_1)
FVq2	D	Rational poly fit, numerator 2 (p_2)
FVq3	D	Rational poly fit, denominator 1 (q_1)
FVq4	D	Rational poly fit, denominator 2 (q_1)
FVq5	D	Rational poly fit, error ($RMSE_r$)

In conclusion, a single 'inclusive' feature vector is used for all the individual classification tasks, generated as the result of data filtering, noise removal, descriptive statistics, generation of the 1st derivative series, encoding via two curve-fitting functions and error-fitting quality metrics. Hence, the pre-processing pipeline acts as a data cleansing and feature generation module that effectively restores the original data in terms of quality, removes outliers and errors, translates the data series into a compact, information-rich vector of the most important inherent characteristics and defines the input space for each subsequent discrimination task.

The pre-processing output and the datasets generated according to the definition in Table I are available for public non-commercial use (CC-NC-SA) [8].

B. Classification models

The second stage of the proposed method consists of using the datasets containing the feature vectors for designing and training various ML architectures and algorithms, ranging from single decision learners to ensembles of classifiers employing voting schemes. Specific models are trained and optimized as such for each task, i.e., olives variety or location of origin.

The feature selection process, as well as the three categories of classifier models employed, are described next.

1) Feature selection: Each classification task, i.e., olives variety or location of origin, employed a separate feature selection process, using exhaustive search for optimal combinations, since the dimensionality of the input space (13 features) is small enough to do so. For the evaluation of each feature subset, two methods have been employed in a complementary sense, namely: (a) CFS-based statistical ranking and (b) classifier-based discrimination ranking. For (a), the Correlation-based Features Subset (CFS) selection method [9] evaluates the gain from each subset of attributes by considering the (statistical) predictive power of each feature compared to redundancies between them. Hence, subsets of features that are highly correlated with the target (class) while having low statistical correlation between them are ranked higher. For (b) the standard Classification and Regression Tree (CART) classifier [10] was used as the discrimination evaluator due to its speed and simplicity. The results from CFS and CART feature selections were then combined together, retaining only the members that were selected by at least one of them (union), thus expecting that smaller combined feature subsets are indeed more robust and descriptive of the domain space.

Table II presents the optimal features subsets used in teh classification tasks, namely Task 1 for olives variety and Task 2 for location of origin for *vK* and *vT* samples, described in detail in section IV-1.

2) Instance-based classifiers: Two types of instance-based or 'lazy' classifiers were included in this study. Specifically, IBkLG [11] is a variation of the typical k-nearest neighbour (k-nn) classifier [10], employing internal cross-validation and log distance weighting. Additionally, K* or 'Kstar' classifier [12] as an improved k-nn alternative, using entropy-based distance function instead of standard geometric distance metric (e.g. Euclidean).

3) Decision trees: The simplicity, fast training and inherent feature selection-like capabilities (top levels) of decision tree

TABLE II
OPTIMAL FEATURE SUBSETS USED IN THE CLASSIFICATION TASKS.

Feature	Task 1	Task 2(*vK*)	Task 2(*vT*)
MK			x
MIN		x	x
MAX		x	x
ENT			x
FVg1			x
FVg2			x
FVg3	x	x	
FVg4		x	x
FVq1	x	x	x
FVq2			
FVq3			
FVq4			x
FVq5	x		

models make them very valuable candidates for feature gain assessment, as well as members of classifier ensembles. Four different types of decision trees were employed in total. Specifically, CART [10] was used in feature selection, as described earlier. J48 tree [13] is a variation of the typical C4.5 decision tree [10], with the ability to employ pruning during training. Reduced Error Pruning (REP) trees [14] are based on iterative construction and pruning that increases information gain or reduces variance, while gradually minimizing the Mean Squared Error (MSE) or other similar error metric. Finally, Random trees [14] employs the Bagging [10] idea to split the initial features set into randomized subsets for training each tree node. The selection of these tree learners, especially the ensembles, is based on the fact that proper data pre-filtering, feature selection, post-training pruning greatly improves any instabilities and noise sensitivity that tree models often show in practice. Furthermore, it has been proven that feature subspace partitioning (Random Forests) or transformation (Rotation Forests) and aggregated decision-making (ensembles) produces at least as efficient learners as any single model of much higher complexity [15], [16].

4) Ensembles: Instead of using a single complex/'strong' model in a classification task, both theory and practice have proven that using a pool of simpler/'weak' models is at least equally effective [15], as well as more efficient in terms of parallelization and training time. Four different ensemble types have been employed in this study. Specifically, the Adaboost M1 algorithm [15], [17] uses the idea of iterative ensemble construction via boosting, i.e., training another separate classifier (J48) for the marginal/error cases from the previous iteration. Multiboost AB is an extension of it [18], employing bifurcation ('wagging') techniques in addition to typical boosting and better robustness to noise-associated errors as a result. Random Forest [15], [19] is e multi-classifier extension of the Random tree approach, training different trees for each randomized feature subset (instead of for each tree node). Finally, Rotation Forest [20] is an improvement over Random Forests, where the original feature space is split into randomized subsets along with rotation transformations, usually via Principal Component Analysis (PCA) [10] or other random projection algorithm, in order to produce not just distinct but also uncorrelated feature subsets. For the final decision of the ensemble, all these approaches employ standard majority voting, which has also been demonstrated as the optimal aggregation scheme for hard-decision tasks [15], [16].

IV. EXPERIMENTS AND RESULTS

1) Experimental protocol: As previously described, there are two classification targets which constitute two distinct tasks, namely (i) Task 1 for olives variety and (ii) Task 2 for location of origin.

For variety (Task 1) the dataset used consisted of 341 samples, prepared according to the pre-processing pipeline described in section III-A, and two target classes, namely 'Koroneiki' (*'vK'*) and 'Tsounati' (*'vT'*), in olive leaves. For the construction of this dataset only one biomarker data were used after the feature selection process according to section III-B1, thus the *MK* feature in Table I was excluded.

For location of origin (Task 2), there are two subsets of the complete dataset, each associated to a single variety, i.e., *vK* or *vT*. This hierarchical approach was selected in order to investigate any bias of better or worse performance of the classifiers on one of the two varieties. Thus, the first subset is for pure *vK* and consists of 268 samples, while the second subset is for pure *vT* and consists of 257 samples, prepared according to the pre-processing pipeline described in section III-A, and both with two target classes, namely 'Chania' (*'rC'*) and 'Rethymno' (*'rR'*), which are nearby geographical regions of Crete, Greece. For the construction of the *vK* dataset only one biomarker data were used after the feature selection process, thus the *MK* feature in Table I was excluded. In contrast, for the *vT* dataset the data from two biomarkers were used after the feature selection process, thus the *MK* feature in Table I was included in the optimal feature subset.

All accuracy rates and generalization capabilities have been estimated with k-fold cross-validation training/testing [10] with $k = 10$, assessing their inherent robustness and real-world performance.[1]

2) Dimensionality reduction: As previously noted, decision tree classifiers provide fast training and explainable structure in the trained models, since decision nodes are based on thresholds for specific features. Furthermore, the selection of specific features in the top levels or repeatedly throughout the tree essentially provides hints of information-rich features, i.e., an intermediate method of highlighting the most important features as in feature selection. The inherent drawback of many decision tree models being sensitive to noise and outliers is remedied via the introduction of tree ensembles, instead of single trees [15].

In the case of domain-transformation models, e.g. with Rotation Forests, the optimal feature subsets are further processed via a projection-approximation transformation like PCA and the resulting 'augmented' feature vectors are used as input

[1] Experimental work was based on: Mathworks MATLAB v9.4/R2018a (x64); Octave v5.1.0; R v3.6.2; WEKA v3.9.4; IBM SPSS Modeler v14.1 & Statistics v26; custom Java & C/C++ tools for data import/export.

for training the ML models. The following is an example illustrating the statistical importance (individual discrimination capacity) of the first three PCA components of the generated features dataset using a single biomarker. These three PCA components are the compact representation of the original 13 features, capable of capturing 98% of their information content (variance). Thus, these are adequate training input for a predictive ML model, capable of discriminating variety in Task 1 (vK-versus-vT), achieving accuracy of more than 96% even with a single J48 or Random tree classifier. It is also a verification for the size of the optimal feature subset for Task 1, as described in Table II.

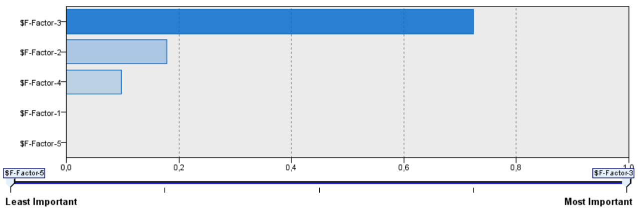

Fig. 5. Example of PCA-based dimensionality reduction of the features space in Task 1.

3) Classification results: Six main classification models were used for comparative results in both tasks, according to their descriptions in section III-B2 and III-B4.

Table III presents the results per (optimized) classification model for Task 1, i.e., olives variety (vK-versus-vT), using the feature subset as noted in Table II. IBkLG was optimized with $k = 4$; all tree ensembles used a pool of size 10 and the J48 as the base classifier, except Random Forest which used Random trees; in Rotation Forest the transformation used was PCA. Due to the saturation of the accuracy rate towards 100%, deviations in all cases were in the order of $\leq 3/341$, i.e., less than $\pm 1\%$ w.r.t. the table values, over several randomization (seed) runs of the k-fold cross-validation process.

TABLE III
TASK 1: OLIVES VARIETY

Classifier models	*Acc%*
Kstar	**96.48**
IBkLG	94.13
AdaboostM1	94.72
MultiboostAB	95.01
Random Forest	94.72
Rotation Forest	94.43

Note: Acc% deviation in all cases is $\leq \pm 1\%$.

Similarly, Table IV presents the results per (optimized) classification model for Task 2, i.e., location of origin (vC-versus-vR) for the two varieties (vK, vT), using the feature subset as noted in Table II. For vK, IBkLG was optimized with $k = 1$; all tree ensembles used a pool of size 10 and the J48 as the base classifier, except Random Forest which used Random trees; in Rotation Forest the transformation used was PCA. For vT the classifiers were configured exactly the same, except from IBkLG which was optimized with $k = 3$. Due to the saturation of the accuracy rate towards 100%, deviations in all cases were in the order of $< 3/341$, i.e., less than $\pm 1\%$ w.r.t the table values, over several randomization (seed) runs of the k-fold cross-validation process.

TABLE IV
TASK 2: LOCATION OF ORIGIN

Classifier models	*Acc% (vK)*	*Acc% (vT)*
Kstar	**99.25**	**99.61**
IBkLG	**99.25**	98.83
AdaboostM1	98.13	98.05
MultiboostAB	97.76	97.66
Random Forest	98.88	99.22
Rotation Forest	98.51	98.44

Note: Acc% deviation in all cases is $\leq \pm 1\%$.

Although the classification results in Tables III and IV refer to very high performance for all models, the underlying problems in both Tasks 1 & 2 are not at all simple or of linearly separable classes. Figure 6 illustrates a X-Y scatter plot with two of the features in the optimal subset (size=5) used in Task 2(vK), i.e., location or origin for the vK samples.

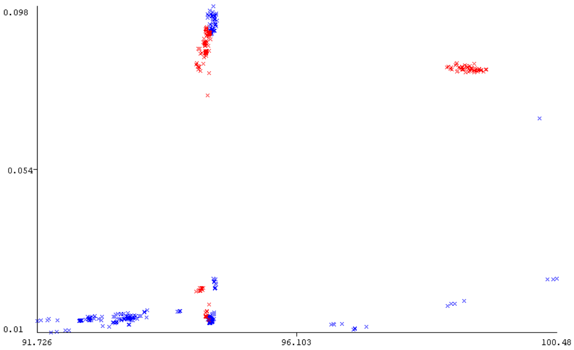

Fig. 6. Geographical discrimination of the same variety (vK) from the regions of Chania (vC=blue) and Rethymno (vR=red); the plot illustrates the two of the most statistically significant features (X=*MAX*, Y=*FVg4*).

V. DISCUSSION AND FUTURE WORK

According to the detailed results in Tables III and IV, the best-performing classifier for both Tasks 1 & 2 is Kstar. For olives variety (Task 1) it achieved accuracy rate of 96.48% over k-fold cross-validation testing. For location of origin (Task 2) it achieved accuracy rate of 99.25% (same as IBkLG) in the vK samples and 99.61% in the vT samples.

It is important to note that, regardless of the classifier model, all experimental tests verify that the pre-processing pipeline and the feature generation process is indeed extremely efficient in two ways. First, in reducing the original input space, from data series of several thousands of points versus temperature ($T \in [70^oC \ldots 95^oC]$, sampled per 0.01^oC) to a feature vector of only 13 values. Second, in capturing the content-rich information from each sample w.r.t. the two classification tasks, i.e., the genetic profiling and 'fingerprinting' for constructing a proper input space for training the models.

Furthermore, this genetic 'fingerprint' is compact enough to be incorporated in various other applications related to highly reliable quality assurance in the food supply chain, protection against fraud, PDO and PGI via 'smart' contracts and blockchain platforms, etc. In practice, any such food product that can be tagged with this DNA-based biometric-like data generator and then tested with this proposed approach is extremely difficult to tamper or degrade, e.g. with admixtures.

Regarding the pre-processing pipeline, it has been proven as robust, reliable and lightweight enough for real-world IoT/edge/MCU applications. Human expert intervention in the list of action points (see section III-A) is necessary only in the design phase, as in step 2. Similarly for step 8, the curve-fitting quality threshold for error can be fixed at an optimal value or even adapted via simple derivative analysis, i.e., sudden transition from sharp drop to almost flat error slopes. All the steps in the pre-processing pipeline, including the feature generation (curve-fitting parameters) and filtering (smoothing and threshold-based), are processes of linear or sub-linear complexity and very low computational requirements.

Specifically for the two curve-fitting steps, each parameter can be easily initialized very close to the expected optimal value, e.g. the Gaussian mean μ at the lower peak of the 1st derivative plot, and any standard optimization method can be employed thereon, since the curve-fitting models include only only three (Gaussian) or four (rational poly) free parameters and relatively smooth underlying data series.

Similarly for the classification models, it is clear from the results that any of the classifiers that were tested was very efficient and robust in terms of performance. Instance-based classifiers have the drawback of exchanging lower complexity for higher storage requirements, as they need a large portion of representative training data available at all times. However, even with a dataset size in the order of 200-300 samples as in Tasks 1 & 2, the feature space is so small (13 at most) that the memory requirements in practice are very low.

On the other hand, having a more complex classifier like Rotation Forest requires more processing per sample even in the recall mode (after training), with the advantage of producing a much more compact trained model. Similarly, the processing requirements in the recall mode are higher especially with the PCA enabled, but again this is fully predetermined during the training phase and at the recall mode it is only a vector rotation with low dimensionality. Therefore, in all cases and choices, the proposed approach remains well within the requirement for lightweight models, applicable to hardware-constrained implementations (IoT/edge/MCU).

VI. Conclusion

In this work, a novel data-driven approach was presented for genetic profiling and 'fingerprinting' of olives and olive leaves via biomarkers and HRM-based DNA analysis, capable of supporting various applications for the food industry including quality assurance, protection against fraud, etc. The approach consists of two stages, namely a pre-processing pipeline for data cleansing and feature generation, and a classification via instance-based or tree ensemble models. Two classification tasks were presented as real-world use cases, regarding the olives variety and the location of origin; both extremely valuable for the food supply chain and relatively unexplored in the context of such applications.

Extensive experimental tests with real-world datasets using various classifiers demonstrated the validity, effectiveness and robustness of the proposed approach, achieving accuracy rates above 96% and 99%, respectively, in the two classification tasks. Furthermore, the proposed solution is based on low-complexity designs, capable of running in hardware-constrained environments and even completely in offline mode (non-networked). It is expected that such approaches of genetic 'fingerprinting' will dominate the food industry in the future, especially in the context of IoT/edge/MCU and 'smart' contracts via blockchain technologies, for provable and tamper-proof authenticity and purity of the food product.

References

[1] S. Fraley, J. Hardick, B. Masek, et.al., "Universal digital high-resolution melt: a novel approach to broad-based profiling of heterogeneous biological samples," Nucleic Acids Research, 41(18):e175, 2013.

[2] P. Athamanolap, V. Parekh, S. Fraley, et.al., "Trainable High Resolution Melt Curve Machine Learning Classifier for Large-Scale Reliable Genotyping of Sequence Variants", PLoS ONE 9(10):e109094, 2014.

[3] H. Li, Alkan Kabakcioglu, "Role of helicity in DNA hairpin folding dynamics," arXiv:1806.08609v1 [q-bio.BM], 2018.

[4] M. Zaboikin, C. Freter, N. Srinivasakumar, "Gaussian decomposition of high-resolution melt curve derivatives for measuring genomeediting efficiency", PLoS ONE 13(1):e0190192, 2018.

[5] J.-H. Jeon, W. Sung, "An effective mesoscopic model of double-stranded DNA," J. Biol. Phys., 40:1–14, 2014.

[6] M. Hosseini-Mazinani, R. Mariotti, B. Torkzaban, M. Sheikh-Hassani, S. Ataei, et.al. "High Genetic Diversity Detected in Olives beyond the Boundaries of the Mediterranean Sea," PLoS ONE, 9(4): e93146.

[7] InTTrust S.A., BIOCOS P.C., "DNA digitization: Olive oil authenticity and traceability from field to bottle (DNAblockchain)," S3FOOD Sub-Grant Agreement N VV106/2021, EU Horizon 2020 (G.A. No 824769).

[8] https://www.inttrust.gr/dnablockchain/data/

[9] M. A. Hall, Correlation-based Feature Subset Selection for Machine Learning. Hamilton, New Zealand: 1998.

[10] S. Theodoridis, K. Koutroumbas, Pattern Recognition (4th/ed.). Cambridge, MA: Academic Press, 2008.

[11] D. Aha, D. Kibler, "Instance-based learning algorithms," Machine Learning, 6:37-66, 1991.

[12] J. Cleary, L. Trigg, "K*: An Instance-based Learner Using an Entropic Distance Measure," Proc. 12th Int. Conf. on Machine Learning, 108-114, 1995.

[13] Ross Quinlan, C4.5: Programs for Machine Learning. San Mateo, CA: Morgan Kaufmann Publishers, 1993.

[14] S. Kalmegh, "Analysis of WEKA Data Mining Algorithm REPTree, Simple Cart and RandomTree for Classification of Indian News," Int. J. Innov. Sci. Eng. Techn. (IJISET), Vol. 2, Issue 2, 2015.

[15] L. Kuncheva, Combining Pattern Classifiers: Methods and Algorithms. New York: Wiley, 2004.

[16] H. Georgiou, M. Mavroforakis, S. Theodoridis, "A Game-Theoretic Approach to Weighted Majority Voting for Combining SVM Classifiers," Int. Conf. on Artif. Neural Networks (ICANN), Part I, LNCS 4131, pp. 284-292, 2006.

[17] Y. Freund, R. Schapire, "Experiments with a new boosting algorithm," Proc. 13th Int. Conf. on Machine Learning, San Francisco, 148-156, 1996.

[18] G. Webb, "MultiBoosting: A Technique for Combining Boosting and Wagging," Machine Learning. 40(2) 2000.

[19] L. Breiman, "Random Forests," Machine Learning. 45(1):5-32, 2001.

[20] J. Rodriguez, L. Kuncheva, C. Alonso, "Rotation Forest: A new classifier ensemble method," IEEE Trans. on Pattern Analysis and Machine Intelligence, 28(10):1619-1630, 2006.