

Food fraud detection using explainable artificial intelligence

Okan Buyuktepe¹  | Cagatay Catal² | Gorkem Kar¹ | Yamine Bouzembrak³ | Hans Marvin⁴ | Anand Gavai⁵

¹Department of Computer Engineering,
Bahcesehir University, Istanbul, Turkey

²Department of Computer Science and
Engineering, Qatar University, Doha, Qatar

³Information Technology Group, Wageningen
University & Research, Wageningen, The
Netherlands

⁴Department of Research, Hayan Group, B.V.,
Rhenen, The Netherlands

⁵Industrial Engineering & Business Information
Systems, University of Twente, Enschede, The
Netherlands

Correspondence

Cagatay Catal, Department of Computer
Science and Engineering, Qatar University,
Doha, Qatar.

Email: ccatal@qu.edu.qa

Abstract

Recently, the global food supply chain has become increasingly complex, and its scalability has grown. From farm to fork, the performance of food-producing systems is influenced by significant changes in the environment, population and economy. These changes may cause an increase in food fraud and safety hazards and hence, harm human health. Adopting artificial intelligence (AI) technology in the food supply chain is one strategy to reduce these hazards. Although the use of AI has been rising in numerous industries, such as precision nutrition, self-driving cars, precision agriculture, precision medicine and food safety, much of what AI systems do is a black box due to its poor explainability. This study covers numerous use cases of food fraud risk prediction using explainable artificial intelligence (XAI) techniques, such as LIME, SHAP and WIT. We aimed to interpret the predictions of a machine learning model with the aid of these technologies. The case study was performed on a food fraud dataset using adulteration/fraud notifications retrieved from the Rapid Alert System for Food and Feed system and economically motivated adulteration database. A deep learning model was built based on this dataset and XAI tools have been investigated on the proposed deep learning model. Both features and shortcomings of the current XAI tools in the food fraud area have been presented.

KEY WORDS

deep learning, explainable artificial intelligence, food safety, interpretable machine learning, machine learning

1 | INTRODUCTION

The world is becoming more interconnected every day and almost anything shipped from a variety of areas throughout the world is purchased by consumers. The globalization of food production, processing and marketing has increased some food hazards, such as infectious diseases (Käferstein et al., 1997). This type of globalization manifests itself in many different industries and involves numerous stakeholders. One crucial industry where its effects cannot be overlooked is the transportation of goods, particularly food. As such, people are generally concerned about food fraud and safety (Nogales et al., 2020). One of the most traded commodities worldwide is food. The global food chain is advancing rapidly and becoming more complex as the world's population grows and markets become more accessible. These factors make food fraud and safety an increasingly important issue.

The availability of sufficient and healthy food is a crucial requirement for maintaining human health (Fukuda, 2015). From farm to fork, every step in the food production process poses a risk to consumer health. This results in the requirement for better defence. The European

This is an open access article under the terms of the [Creative Commons Attribution-NonCommercial-NoDerivs](#) License, which permits use and distribution in any medium, provided the original work is properly cited, the use is non-commercial and no modifications or adaptations are made.

© 2023 The Authors. *Expert Systems* published by John Wiley & Sons Ltd.

Commission declared that rigorous food safety rules are their top priority in response to these worries (Mutukumira & Jukes, 2003). The European Food Safety Authority (EFSA) collects scientific data and offers unbiased scientific advice on food-related risks in order to carry out this policy. The Rapid Alert System for Food and Feed (RASFF) is made up of the EFSA, national food safety agencies from EU Member States, the European Commission, Norway, Liechtenstein, Iceland and Switzerland. The RASFF facilitates information sharing among its members and offers a service to guarantee that urgent notifications are handled properly to prevent problems with food safety. Additionally, RASFF has an online platform called the RASFF Portal that makes it easier to register incidents of health problems when they are discovered. Manual approaches to identifying these hazards take a lot of effort and are prone to mistakes. As a result, automated smart techniques must be used. AI techniques can be used to foresee these risks and significantly lessen their impact. In this study, we built a predictive model powered by AI and ran our experiments using a dataset generated from notifications entered into the RASFF system.

Machine learning (ML), a branch of artificial intelligence, enables software to predict outcomes more accurately without being explicitly programmed. ML algorithms predict new output values using historical data. Deep learning (DL), a branch of ML, uses algorithms modelled based on biological neurons. Powerful processors, enormous datasets and adaptable software libraries have helped make DL one of the most widely used ML approaches today. The definition of DL is models with numerous processing layers that develop a representation of the data with various degrees of abstraction (Le Cun et al., 2015). Modern AI-powered systems provide cutting-edge solutions to a wide range of challenging issues in areas like health care, nutrition, agriculture, energy and transportation that impact people's lives. These systems require low human involvement, and their error margin is small. Recently, DL algorithms have achieved the highest accuracy performance for challenging problems including face recognition, object identification and image segmentation. Although these algorithms provide incredibly precise responses, it is sometimes challenging for people to understand how the machine arrived at that conclusion. Because of this, AI researchers have put a lot of effort into developing tools, processes and strategies that allow people to understand and trust the results of ML-based models. This research area is called explainable artificial intelligence (XAI). It is used to describe the reasons behind and mechanisms underlying the biases of an AI-based model (Arrieta et al., 2020).

It has become necessary to comprehend how AI-based models in such systems make decisions (Goodman & Flaxman, 2017). Knowing why a model generates a particular prediction may be just as important as understanding how accurate its forecasts are. Even experts, engineers and data scientists find it challenging to evaluate complex models like ensemble models (like multiple classifier systems) or DL (Lundberg & Lee, 2017). The field of XAI includes a number of techniques and procedures that let users of a ML-based model trust and understand their outputs. It is generally accepted that improving our understanding of a system can aid in resolving any potential issues it may have. As a result, XAI is crucial to the implementation of AI-based models. While production rules-based, simpler AI systems, such as those that use if-then-else statements, can be clearly understood, deep neural networks (DNNs)-based models are too complex to be easily explained (Arrieta et al., 2020). DNNs are criticized for being black-box models because they have millions of parameters and multilayer nonlinear architectures (Castelvecchi, 2016). The interpretability of DL models is essential due to the large range of applications they are used in. Assuring the model's fairness, enhancing its resilience and guaranteeing that significant inputs infer the conclusion are three advantages of treating interpretability as a separate design consideration (Arrieta et al., 2020).

The goal of this research is to develop a DL model that predicts the type of food fraud in order to increase the interpretability of the suggested DL model in light of the aforementioned challenges. Marvin and colleagues published the data utilized for this article in a previous study (Bouzembrak & Marvin, 2015) that used a probabilistic Bayesian Network model. Because the model did not use DL approaches, the previous study did not concentrate on the models' explainability. In this research, XAI tools had to be looked at because we wanted to boost the model's overall performance and address the explainability issues.

The contributions of this study are as follows:

1. This study is the first to use XAI methods on a real-world dataset related to the field of food fraud and safety.
2. The XAI tools' weaknesses and advantages have been emphasized. This will spur more study in this area and aid in the development of original XAI tools for the food safety domain.
3. Professionals in food fraud and safety can benefit from the findings of this study to enhance the food supply chain.

The following sections are arranged as follows: Section 2 provides background information and related work. The dataset's characteristics are explained in Section 3, along with the methods and techniques used for pre-processing and building DNNs. The XAI tools are explained in Section 4. Section 5 presents the results and Section 6 shows the discussion. The conclusion and future work are presented in Section 7.

2 | BACKGROUND AND RELATED WORK

This section presents DL tools first, followed by XAI tools.

2.1 | Deep learning

A part of ML known as DL is a neural network model. Most of the important features are learned using a DL technique via the hidden layer architecture. Each node in the network represents a different aspect of the object of interest, and when those facets are combined, they make up the entire thing. The strength of each node's connection to the object can be measured using the weights that have been given to each node. The weights are adjusted when the model changes across various iterations.

The fact that the DL algorithms need a huge number of data points is one of their key characteristics. As a result, experiments to determine the optimal weights, which take place throughout the training of the network model, require a significant amount of processing resources. The accuracy of findings from DL algorithms can vary depending on the dataset. However, because these models are black-box models by design, the end user cannot immediately see the connection between the input and the output. It is necessary to investigate the explainability of these hypotheses using a variety of instruments, methodologies and approaches.

2.2 | Explainable artificial intelligence tools

An emerging field called XAI offers a number of methods for transforming the opaque nature of models based on ML or DL and creating explanations that are understandable to humans. Researchers in AI and ML are paying more attention to XAI as a result of the considerable advancement in the application of ML and DL techniques. To comprehend the black-box models, researchers have created a variety of tools. Local Interpretable Model-Agnostic Explanations (LIME) (Ribeiro et al., 2016) creates explanations regarding feature contributions to the creation of a prediction using any ML model as input. The model's output is condensed in Shapley Additive exPlanations (SHAP) (Lundberg & Lee, 2017), which also identifies the key characteristics that influence the model's choices. A model-specific approach to discussing DL models is Deep Learning Important FeaTures (DeepLIFT) (Shrikumar et al., 2017). A program called ELI5 (TeamHG-Memex, 2022) is used to demonstrate several Python implementations of ML models. Skater (Skater, 2022) offers a method for comprehending the various ML libraries' learning frameworks. Understanding the choices made by ML models is made easier by Machine Learning extensions (MLxtend) (Raschka, 2018). Microsoft's given ML models are better-understood thanks to InterpretML (Kaur et al., 2020). Alibi (Getting Started, 2022) explains and provides information on ML models. What-if Practitioners can investigate, visualize and research ML systems using the tool (WIT) (Wexler et al., 2019). In this work, we used LIME, SHAP and WIT tools to forecast food fraud and observed their outputs. Significantly more studies used the SHAP and LIME tools (Aldughayfiq et al., 2023; Dieber & Kirrane, 2022; Kuzlu et al., 2020; Lombardi et al., 2021; Parsa et al., 2020; Sahay et al., 2021; Wenbo et al., 2018). Additionally, a large technology company and undergraduate students investigated the WIT tool for testing purposes; some significant results were found using this tool (Wexler et al., 2019). These tools are chosen for use in this research as a consequence. These three tools have been chosen based on their potential, adaptability and suitability for this situation. We want to see the features that affect the DL model's prediction, the behaviour of the model depending on these features, and the characteristics of the dataset using the data provided by these tools so that we may build a better DL model for early warning and prediction of food safety issues.

2.2.1 | Local Interpretable Model-Agnostic Explanations

By roughly modelling any black box ML model for elucidating each prediction, LIME is an approach that offers a local interpretable model (Ribeiro et al., 2016). This technique modifies the initial data points and then gets the associated predictions by feeding the underlying model the new data points. Finally, LIME trains an understandable model using these new data points and weights them based on how close they are to the original points. Then, each original data point may be explained using this explanation model. In Equation (1), G represents the family of potentially interpretable models, and g is the explanation model of instance x where $g \in G$ and $\Omega(g)$ is the complexity of the interpretable model g . The proximity measure π_x defines how large the neighbourhood around instances x and f is the complex model being explained.

The explanation model g minimizes $L(f, g, \pi_x)$, which measures how close the explanation is to the prediction of the original model f while having complexity $\Omega(g)$ low enough to be interpretable.

$$\xi(x) = \arg \min_{g \in G} L(f, g, \pi_x) + \Omega(g) \quad (1)$$

2.2.2 | SHapley Additive Explanation

The game theory led to the development of the Shapley Value, which measures a player's marginal contribution (Shapley, 1953). Simply using Shapley values is how the explainable AI tool SHAP operates. These numbers define the average feature contribution to the prediction (Christoph, 2020). SHAP is defined in Equation (2),

$$g(z') = \phi_0 + \sum_{j=1}^M \phi_j z'_j \quad (2)$$

where g is the interpretation model, $z' \in \{0,1\}^M$ shows whether the corresponding feature can be observed (1 or 0). $\phi_j \in \mathbb{R}$ is the attribute value (Shapley value) of each feature and M is the number of inputs. The basic idea of Shapley values is to calculate a player's contribution for each subset and then, simply average over all these contributions (Christoph, 2020). SHAP offers global and local interpretability and also suggests a model-agnostic approximation for SHAP values.

The collective SHAP values are used to demonstrate how much each predictor contributes, either positively or negatively, to the target variable. The importance of a feature j is defined by the SHAP as shown in Equation (3):

$$\phi_j = \sum_{S \subseteq Z \setminus \{j\}} \frac{|S|!(M - |S| - 1)!}{M!} [f_x(S \cup \{j\}) - f_x(S)] \quad (3)$$

where $f_x(S)$ is the output of the model to be interpreted using a set of S features. N denotes the complete set of all features. The contribution of feature j (ϕ_j) is determined as the average of its contribution among all possible permutations of the feature set.

2.2.3 | What if Tool

A visual interface called the What-if Tool (WIT) makes it easier to comprehend any dataset and the results of ML models that operate in a black box (Wexler et al., 2019). The PAIR (People + AI Research) initiative released WIT. With the help of this tool, trained ML models may be tested effectively, simply and without writing any code. Data point editor, performance and features overview are the three tabs that make up the WIT interface. WIT offers a number of benefits, including the ability to compare multiple models within the same workflow, edit a data point to see how the ML model performs, contrast the data points with counterfactuals, visualize inference outputs, arrange the data points by similarity, and test algorithmic fairness constraints.

3 | MATERIALS AND METHODS

The dataset and procedures utilized in this study are described in this section. This dataset was created for the authors' prior paper (Bouzembrak & Marvin, 2015), which used Bayesian networks to address the issue of food fraud.

3.1 | Data description

Two numerical features and 27 categorical features make up the dataset. Cases of food fraud were gathered from the RASFF and EMA databases. The occurrences of food fraud were chosen from the time frame of 1 January 2000 to 31 December 2015. Seven different categories were used to examine the cases of food fraud that were reported to the EMA and RASFF databases. Consequently, the dataset contains seven different fraud types, as illustrated in Table 1.

Four food fraud specialists, the EMA and RASFF databases, and finally some brainstorming techniques are used to identify the features (Bouzembrak & Marvin, 2015). The dataset includes features such as the year, month and data source of the food fraud case, as well as the name, category, price in year and month, and country of origin of the product. There are also some missing values in the dataset. These values ought to have either been imputed or eliminated before using the DL approach. A Python library called DataWig (Biessmann et al., 2019) aids in the imputation of missing values for categorical characteristics in datasets. DataWig combines autonomous hyperparameter tweaking and DL feature extractors. It is a reliable and scalable method for the imputation of missing values that may be used with tables that contain a variety of data kinds. The StandartScaler imputation algorithm (ScikitLearn Documentation, n.d.) was used to impute the year and month using numeric features. This technique substitutes a new value computed by the technique for the missing values.

3.2 | Methods

Following generic pre-processing, one-hot encoding is used to encode categorical variables so that ML techniques can be used. The majority of ML models only function with numerical variables. It is intended that all inputs and outputs will be numerical. As a result, in order to convert

TABLE 1 Food fraud types.

Fraud type	Description
Fraud type 1: Health certificate (HC)	Improper, fraudulent, missing or absent HC
Fraud type 2: Illegal importation	Illegal or unauthorized import, trade or transit
Fraud type 3: Tampering	Adulteration, fraud or tampering, substitution, counterfeit, artificial enhancement, transhipment, intentional distribution of contaminated product, dilution.
Fraud type 4: Common entry document (CED)	Improper, expired, fraudulent or missing CED, import declaration or analytical report
Fraud type 5: Expiration date	Expiration date
Fraud type 6: Origin labelling	Mislabelling, origin labelling
Fraud type 7: Theft and resale	Theft and resale

Note: This table is adapted from (Bouzembrak & Marvin, 2015).

TABLE 2 One hot encoding example for Product_category variables.

Original data		Transformed data		
Product_category		Product_category_Fish	Product_category_Meat	Product_category_Nuts
0	Fish	1	0	0
1	Meat	0	1	0
2	Nuts	0	0	1

categorical features into numeric features, all categorical feature variables in the dataset must be encoded. Integer encoding and binary encoding are two examples of encoding methods. The one-hot encoding approach is the most used strategy for encoding categorical information. According to this method, each category is converted into a vector that contains the numbers 1 and 0, signifying whether the feature is present or not. During the dataset pre-processing, one hot encoding was used. Table 2 provides an illustration of one.

Many ML methods work better with characteristics that are roughly equal in size and near to having a normal distribution. When features are approximately the same size across the dataset, they become equally essential and simpler for most ML models to process. The StandardScaler method of the ML toolkit scikit-learn is used to scale the two numerical features of the dataset. Through the removal of the mean and scaling to unit variance, this approach standardizes a feature.

In this paper, we developed a DNN model made up of numerous perceptron layers. It is also known as a class of feed-forward supervised neural networks known as multi-layer perceptron (MLP). Input, hidden and output layers are the three different kinds of layers found in MLP. It has an arbitrary number of hidden levels in addition to a single output layer and a single input layer. The network's neurons activate nonlinearly using methods like sigmoid and rectified linear units (ReLU). The DNN algorithm was selected in this research because it is capable of learning multiple levels of input representations, it can handle complex problems, and it is flexible to be applied to a wide range of tasks. Also, the learning curve is not steep.

To improve the model's accuracy, DNNs include a few hyperparameters that can be adjusted. Grid search is one technique for hyperparameter tuning that identifies a model's optimal hyperparameters. There are also several other techniques, such as random search, genetic algorithms, gradient-based methods and simulated annealing to optimize the hyperparameters.

Seven neurons are used to construct the output layer of a DNN model since our multi-class dataset contains seven different types of food fraud. This output layer's activation function, Softmax, is chosen to give each output neuron a value between 0 and 1. This number is a likelihood score for the network's right output. In the output layer, the total of all the neurons' outputs equals 1. Given that the challenge is a multi-class classification task, the categorical cross-entropy is chosen as a loss function. Keras, a high-level API built on top of the open-source library TensorFlow, is used to create MLP models. The factors that affected to select Keras framework are as follows:

1. It is easy to use and helps for fast experimentation without the need for low-level code.
2. It is powerful and scalable because it is built on top of TensorFlow.
3. Commonly used algorithms are available within the framework, which shortens the implementation time.

Table 3 displays the model's final configuration as well as the evaluated hyperparameters.

TABLE 3 GridSearch hyperparameters.

Hyperparameters	Values	Selected value
Number of hidden layers	2, 3, 4	3
Neurons per layer	32, 64, 128, 220, 310, 320	320, 64, 32
Activation function	ReLU	ReLU
Dropout	0.0, 0.1, 0.01, 0.001	0.0
Epoch	10, 50, 100, 150	50
Optimizer	Adam	Adam
Learning rate	0.1, 0.01, 0.001	0.01

```
[ ] # Create the LIME Explainer
explainer = lime.lime_tabular.LimeTabularExplainer(train_data.values,
                                                    feature_names = feature_names,
                                                    class_names=class_names,
                                                    categorical_features=categorical_features,
                                                    categorical_names=categorical_names,
                                                    mode="classification",
                                                    kernel_width=3)
```

FIGURE 1 The creation of LIME explainer.

4 | IMPLEMENTATION OF XAI TOOLS

In this section, the implementation of XAI tools is examined.

4.1 | LIME

First, the LIME package was installed. A function is created to return the target variable's expected probability given the set of attributes. The next step is to list the feature names. The parameters used to generate a LIME explanation are as follows:

- Training data values
- All feature names
- Class names—the target values
- Categorical features—the list of categorical columns
- Categorical features—the list of categorical column names
- Mode—the prediction mode—classification
- Kernel width—a parameter that controls the linearity of the induced model; the wider the width, the more linear the model.

Figure 1 shows how to create the LIME Explainer.

The explanations for the stated values in the test dataset are then obtained from LIME as the last step. We select particular observations from the test dataset in order to extract the probability values for each class. LIME will give the initial justification for the probability assignment. We can compare the probability values for that prediction to the target variable's actual class. To create an explanation figure, the following criteria may be used:

1. An instance picked from the test dataset.
2. The function the returns predicted probability for the target variable.
3. The number of features desired to be displayed in the figure.
4. The number of top predictions to be shown specifies the top class(es) with the highest probability from the prediction.

5. The number of target features desired to display prediction probability values.

Figure 2 shows how to create the LIME Explanation figure.

4.2 | SHAP

We began by installing the SHAP package. A function is created to return the target variable's expected probability given the set of attributes. To estimate SHAP values for any model using a specifically weighted local linear regression, we used Kernel SHAP. Train data and the function are used to build the Kernel Explainer. By providing the test dataset and the number of samples as parameters to the algorithm shown in Figure 3, we calculated the Shapley values.

The Explainer uses a number of techniques to generate explanation figures. For instance, the Summary Plot can be made in the way depicted in Figure 4. A list of the SHAP charts shown in Figure 4 is provided as follows:

1. *Summary plot*: Based on shape values taken from the test dataset, we created a summary plot with a bar plot type. The average impact of each attribute on the outcome of the prediction is shown by the bar chart. The relevance of the features in this graphic is also highlighted by the Shapley values. To create a Summary Plot, the following variables can be used:

```
[ ] # Choose the observation in the test dataset for which explanation is needed and get the explanation
instance = 12
explanation = explainer.explain_instance(test_data.iloc[instance], predict_fn_lime, num_features=10, top_labels=1, labels=range(7))
explanation.show_in_notebook(show_all=True, show_table=True)
```

FIGURE 2 The creation of LIME explanation figure.

```
[ ] # Use Kernel SHAP to explain test set predictions
explainer = shap.KernelExplainer(pred_fnc_shap, train_data)

# Calculate the Shapley values
shap_values = explainer.shap_values(test_data, nsamples=200)
```

FIGURE 3 SHAP explainer creation and Shapley values calculation.

```
[ ] # Summary Plot
shap.summary_plot(shap_values, test_data, plot_type='bar', class_names= class_names)

# Plot the SHAP values for the Tampering output of the 12.th instance :

## Force Plot
plot = shap.force_plot(explainer.expected_value[5],
                      shap_values[5][12],
                      test_data.values[12],
                      feature_names = feature_names,
                      out_names='Tampering')

## Waterfall Plot
shap.waterfall_plot(shap.Explanation(values=shap_values[5][12],
                                       base_values=explainer.expected_value[5],
                                       data=test_data.iloc[12],
                                       feature_names=feature_names))

## Decision Plot
shap.multioutput_decision_plot(explainer.expected_value,
                               shap_values,
                               row_index=12,
                               feature_names=feature_names,
                               highlight = [1])
```

FIGURE 4 Creation of SHAP explanation.

- Shapley values
 - Test dataset values
 - Plot type–bar
 - Class names–target features
2. *Force plot*: It presents Shapley values using an additive force architecture. It might help us identify the elements that affected prediction the most favourably or unfavourably. A Force Plot can be created by using the following variables:
- Base value—the average model output across the training dataset we supplied.
 - Shapley value of the particular instance
 - Test data set value of the particular instance
 - Feature names
 - Target feature
3. *Waterfall plot*: It discusses a specific Shapley value-based model prediction. The Shapley values that were added to the base value to get a particular forecast are shown in this figure. A waterfall plot can be created by using the following variables:
- Shapley value of the particular instance
 - Base value
 - The test dataset's value for a particular instance
 - Feature names
4. *Decision plot*: It shows how the model made a particular decision based on the shape values of several features. To create a decision plot, the following variables can be used:
- Base value
 - Shapley values
 - The test dataset's value for a particular instance
 - Feature names
 - Highlight—A list of indices describing the samples to highlight from the collection of samples

4.3 | WIT

The WIT tool is initially installed in order to use the WIT package. A function is created to return the target variable's expected probability given the set of attributes. These variables are used to generate a WIT visualization user interface:

1. Train or test the dataset itself and target features.
2. The function that returns predicted probability for the target variable (optional).

Additionally, we employed techniques for designating the target feature, the target feature column and the kind of the model as multi-class classification. Figure 5 shows how to create the WIT visualization.

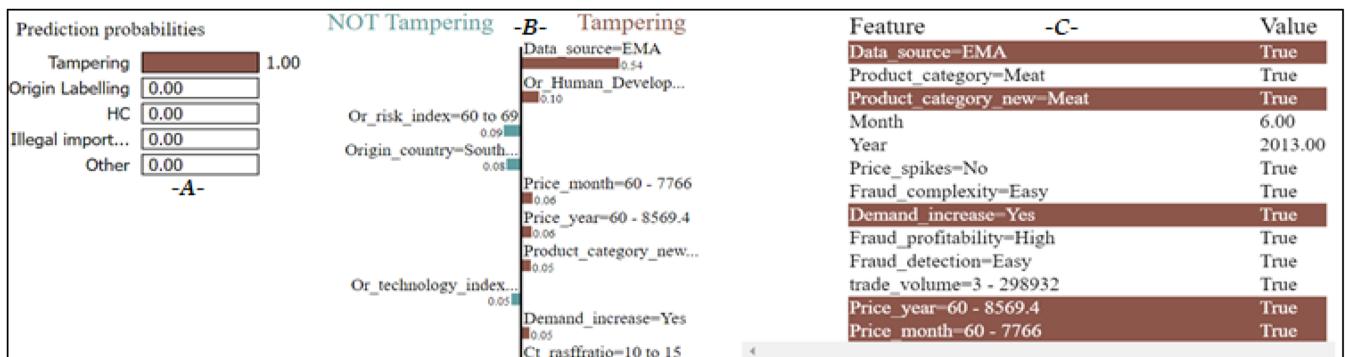
5 | RESULTS

The WIT tool can be used to acquire the DL model output. The accuracy of the model is 81.4%. Additionally, the tool provides the confusion matrix with a seven-by-seven table (number of predicted classes) that lists whether the model's predictions were accurate or not. The predictions of the model are shown on one axis, and the actual data is shown on the other. In Figure 6, the confusion matrix is depicted. The tool shows a receiver operating characteristic (ROC) curve that contrasts the true positive rate (TPR) and the false positive rate at various classification levels (FPR). A precision-recall (PR) curve that displays precision versus recall at various categorization levels can also be produced. The Appendix section includes images of the ROC and PR curves.

```
[ ] # Create a What-if Tool visualization
config_builder = (WitConfigBuilder(test_examples.tolist(), test_data.columns.tolist() + ['Product_fraudtype'])
    .set_multi_class(True)
    .set_custom_predict_fn(predict_fn)
    .set_target_feature('Product_fraudtype')
    .set_label_vocab(['CED', 'Expiration date', 'HC', 'Illegal importation', 'Origin Labelling', 'Tampering', 'Theft and Resale']))
WitWidget(config_builder, height=800)
```

FIGURE 5 WIT visualization creation.

Confusion Matrix											Total
	Predicted 0	Predicted 1	Predicted 2	Predicted 3	Predicted 4	Predicted 5	Predicted 6				
Actual 0	2.9% (8)	0.0% (0)	1.4% (4)	0.7% (2)	0.0% (0)	0.0% (0)	0.0% (0)	5.0%	(14)		
Actual 1	0.0% (0)	1.4% (4)	0.4% (1)	0.4% (1)	0.0% (0)	0.4% (1)	0.0% (0)	2.5%	(7)		
Actual 2	1.8% (5)	0.0% (0)	20.0% (56)	2.9% (8)	0.0% (0)	0.4% (1)	0.0% (0)	25.0%	(70)		
Actual 3	0.7% (2)	0.0% (0)	1.8% (5)	13.6% (38)	0.0% (0)	0.4% (1)	0.0% (0)	16.4%	(46)		
Actual 4	0.0% (0)	0.0% (0)	0.4% (1)	0.4% (1)	10.7% (30)	3.2% (9)	0.4% (1)	15.0%	(42)		
Actual 5	0.0% (0)	0.0% (0)	0.0% (0)	1.1% (3)	1.1% (3)	32.5% (91)	0.0% (0)	34.6%	(97)		
Actual 6	0.0% (0)	0.0% (0)	0.0% (0)	0.0% (0)	0.0% (0)	1.1% (3)	0.4% (1)	1.4%	(4)		
Total	5.4% (15)	1.4% (4)	23.9% (67)	18.9% (53)	11.8% (33)	37.9% (106)	0.7% (2)				

FIGURE 6 Confusion matrix.**FIGURE 7** LIME explanation of instance 12.

5.1 | Applying LIME XAI tool to food fraud prediction

Instead of the global surrogate model, LIME concentrates on the local interpretable model. Figures 7, 8, 9, and 10 show the outcomes of three different test data examples' local explanations. LIME creates a notebook format with three parts that contain all of the results of a certain feature. The original feature values are displayed in the right-hand part (C), while the middle part (A) displays the LIME explanations of selected characteristics. The left part (A) displays the Product Fraud Type prediction probabilities. The findings show that tampering, expiration date and origin labelling were correctly predicted as product fraud types. Tampering is the projected product fraud type for instance 12, as depicted in Figure 7. Data source: EMA; press index: 20.00–29.9; human development index: 0.6000–0.6999; product category: Meat; and Price year: 60–8568.4 have favourable effects; Origin country: South Africa, risk index: 69–69, and technology index: 30.0–39.9 have adverse effects.

In Figure 8, the product fraud type is predicted as the expiration date. Origin country = Belgium; data source = RASFF; press index = 0–9.9; transparency index = 50–59; and product name = Cereals in orange have positive impacts on the prediction.

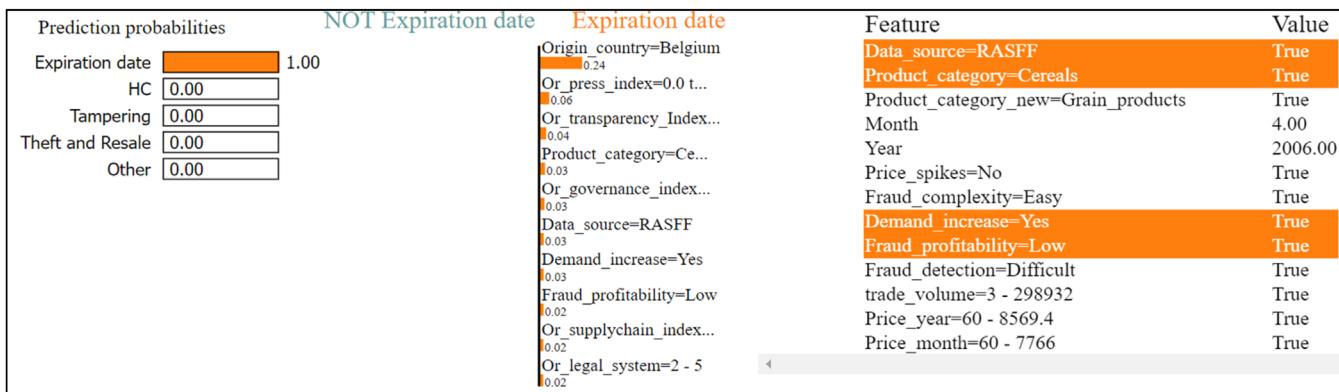


FIGURE 8 LIME explanation of instance 15.

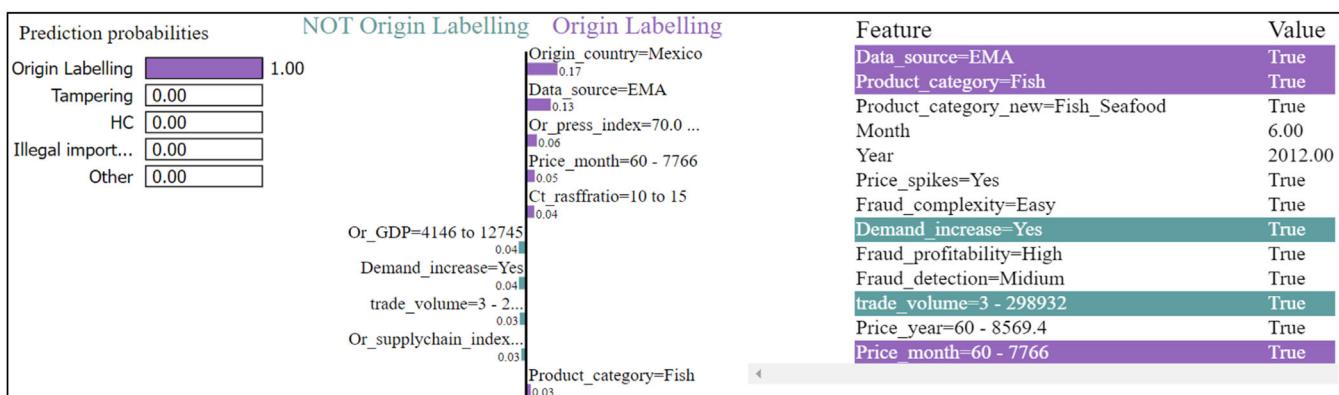


FIGURE 9 LIME explanation of instance 16.

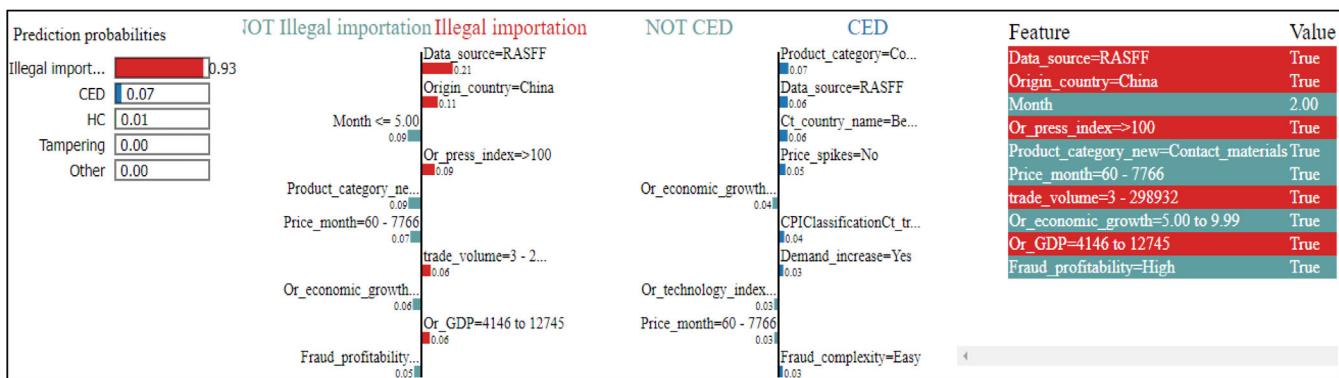


FIGURE 10 LIME explanation of instance 98.

Figure 9 shows that product fraud type is predicted as origin labelling. Origin country = Mexico; data source = EMA; press index = 70.0–79.9; and product name = Fish in purple have positive impacts while GDP = 4146–12,745; demand increase = Yes; and trade volume = 3–298,932 in grey have a negative impact on the prediction.

In the vicinity of the data point of interest, the LIME tool builds a basic interpretable model that approximates the complicated model locally. The incorrectly anticipated instance is shown in Figure 10. Although CED is the actual type of product fraud, illegal importation has a higher prediction probability value according to the model. Additionally, LIME enables us to view the prediction probabilities for all labels; Figure 10 shows two product fraud types.

5.2 | Applying Shap XAI tool to food safety prediction

Based on determined Shapley values, SHAP offers both global and local interpretations. The number of features in a dataset affects how long it takes to compute Shapley values. The time required to calculate values grows exponentially as the number of features rises. These numbers represent computed permutations of all feasible dataset subsets. As a result, it takes a lot of computing power to calculate all those permutations of the subset. The output of a model's SHAP values explains how features affect the model's output. Each Shapley value in this study provides details on the impact of each feature—positive or negative—on the classification of food fraud. The summary figure in Figure 11 groups the features according to how important they are to the model. The graph demonstrates that the feature data source has a greater impact on food fraud type tampering prediction than HC. The most influential features are also the data source, origin country and product name.

Additionally, SHAP offers force plot local interpretability of the models for each data point. The force plots in Figures 12, 13 and 14 were produced with the SHAP tool using the same instances that were previously utilized with LIME. The outcomes show how easily one model prediction may be explained. The model score is pushed higher by characteristics with red colouring, and lower by features with blue colouring. Additionally, the wider the range, the greater the impact on a feature's prediction. Figure 12 shows that data source = EMA enhances the value of the property while risk index = 60–69 decreases the value of the property.

In Figure 13, data source = RASFF and origin country = Belgium, fraud profitability = Low increase the property values.

In Figure 14, data source = EMA, origin country = Mexico increase the property values while supply chain index = 3.50–3.99 reduces the property value.

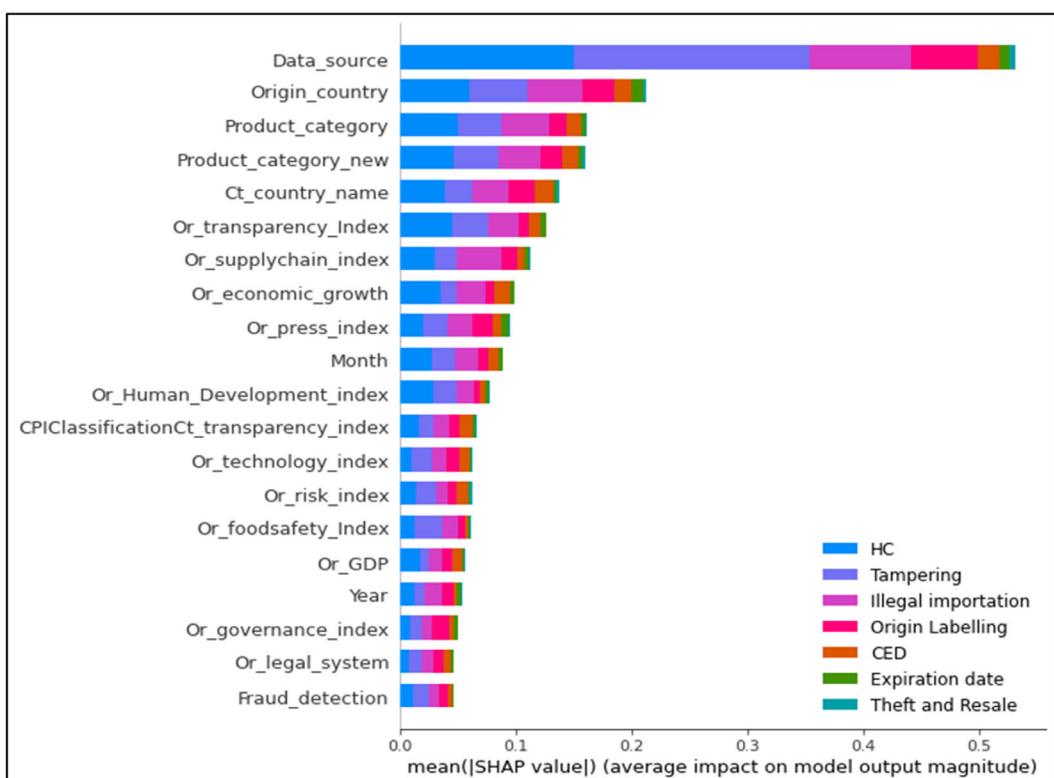


FIGURE 11 SHAP summary plot.

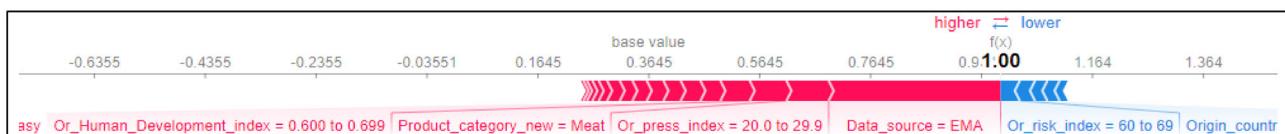


FIGURE 12 SHAP explanation for instance 12.



FIGURE 13 SHAP explanation for the instance 15.

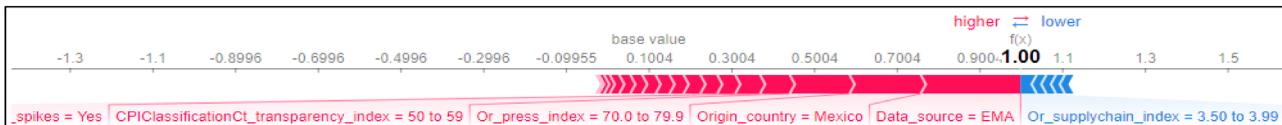


FIGURE 14 SHAP explanation for the instance 16.

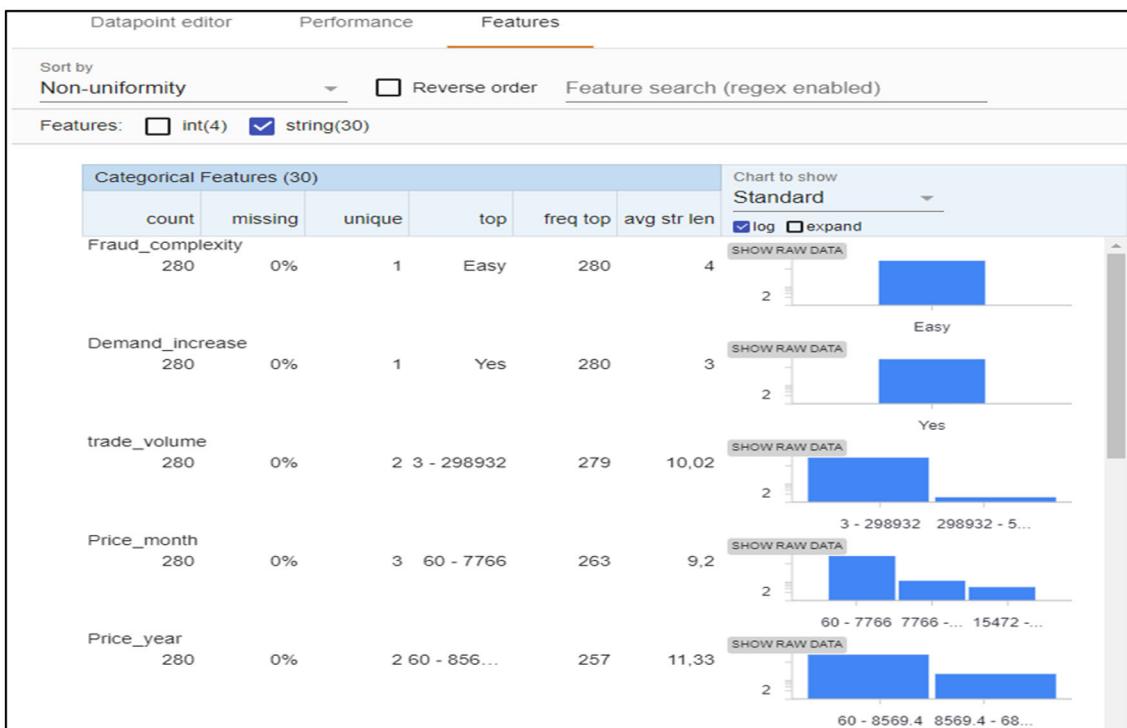


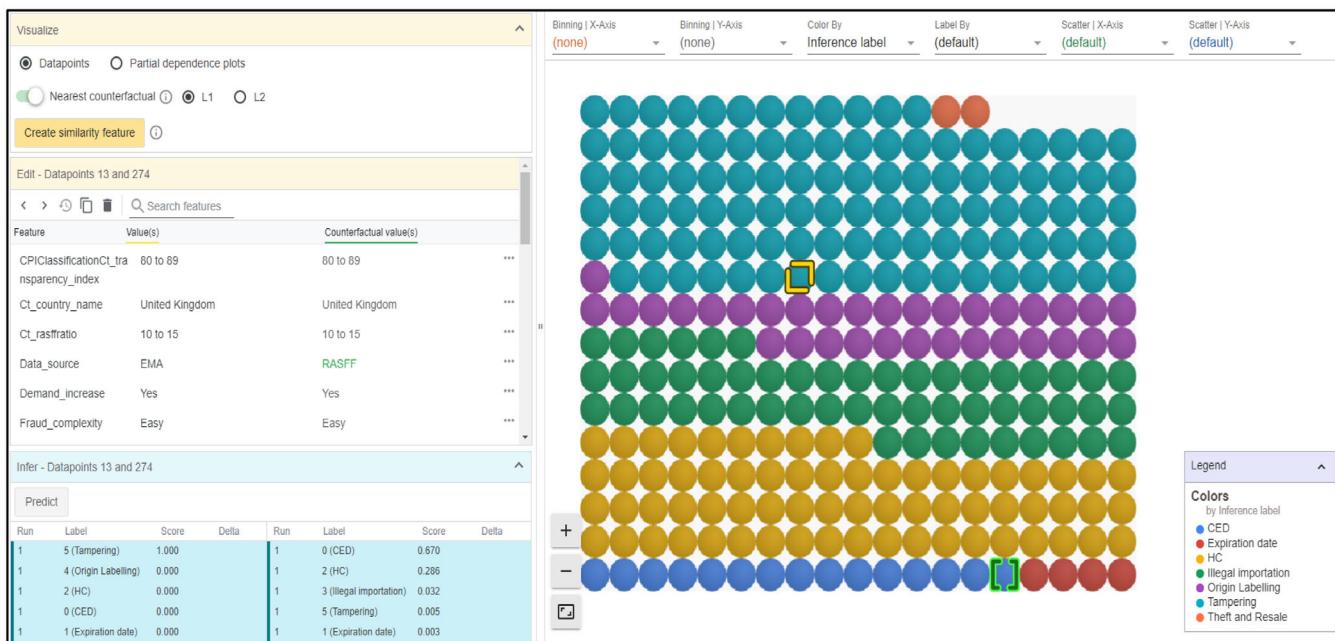
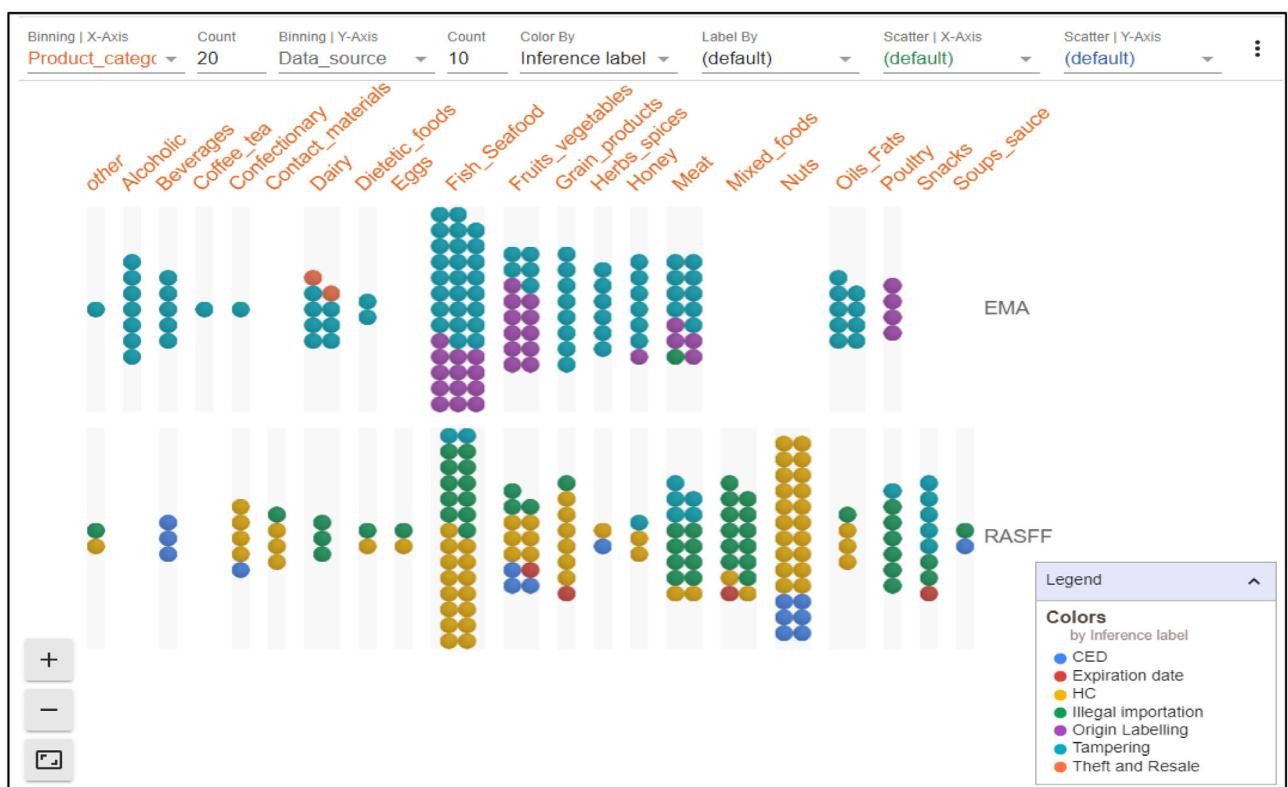
FIGURE 15 WIT features tab.

5.3 | Applying WIT XAI tool to food fraud prediction

In the features tab of WIT, we can view the distribution of values for each feature in the dataset. Figure 15 shows the features of an imbalanced distribution by sorting the data points according to non-uniformity. The figure shows that fraud complexity has a unique value of Easy, indicating that it is not distributed equally. Demand growth is also not uniform. Additionally, the distribution of trade volume, price month, price year and RASFF ratio is not uniform.

The data point editor tab can be used to better understand how the model behaves. We can see how the model modifies its choice by introducing a series of modifications, or “counterfactuals”. The data point with the most comparable feature values and the opposite prediction is shown here. Figure 16 shows the chosen case (in turquoise) and its counterfactual datapoint (in blue). The closest counterfactual examples are the same, but the values for the product name, fraud profitability and data source are different.

The 2D histogram in Figure 17 shows that the Product name = Fish–Seafood is mostly labelled tampering and origin labelling as food fraud type in data source–EMA, while is mostly labelled HC as food fraud type in data source–RASFF.

**FIGURE 16** WIT data point editor tab.**FIGURE 17** WIT data point editor (Table 2).

5.4 | Observations

Three different XAI tool types—LIME, SHAP and WIT—were examined for the food fraud domain in this work, and several observations were made. For example, computation time, code complexity, usability and explainability type aspects were considered. It is determined that each of

TABLE 4 Execution time of the XAI tools.

Model	Time
WIT	4.58 s
SHAP	87.37 min
LIME	2.54 s

TABLE 5 Comparison XAI tools.

	Model		
	WIT	SHAP	LIME
Execution time	Fast	Slow	Normal
Explanation type	Locally/globally	Locally/globally	Locally
Implementation	Easy	Hard	Normal
Graphical variety	Many	Many	Only one
Extract findings	Required manual actions	Automatically	Automatically
User interface	Yes	No	No
Runtime requirements	Dataset + model (optional)	Dataset + model	Dataset + model
Model testing capability	Yes	No	No

these instruments has its own restrictions, benefits and drawbacks. All of these tools offer binary/multi-class classification models as well as regression. LIME can only offer a local explanation for each data point. LIME is model-agnostic; it may be used with any ML model. LIME perturbs data points to produce a local surrogate model. As a result, the local explanation result plot of a single data point may vary depending on the execution in order to recreate the LIME's plot. A local explanation strategy is SHAP. It also gives good all-around justifications. Shapley value computation is computationally intensive. LIME and WIT are hence quicker than SHAP. Results and charts from SHAP can be more in-depth. Examples include summary plots, dependence plots and force plots. In comparison to other tools, SHAP is not the easiest to use. WIT only works with datasets and only a few lines of code. WIT is unable to generate any global or local explanation on its own. Results should be manually disclosed. WIT has a very user-friendly user interface. Some of this tool's advantages include testing the model and displaying the dataset. We also include the execution times for each XAI tool in Table 4. WIT and LIME were quite quick, but SHAP required the most time to execute. Additionally, their execution times are proportionate to the tools' functionality. In Table 5, we also compared XAI tools based on several dimensions.

6 | DISCUSSION

Food fraud and safety have gotten more difficult, according to a number of publications, and there are a number of problems that need to be resolved. For instance, it is necessary to adapt the intricate supply chain to changing conditions. To the greatest extent possible, AI technology should be used to stop food fraud. In order to get further insights, the current study created a DNN model and used XAI technologies. One conclusion from the analysis is that the feature data source is significant among others, either in a globally or locally explicative sense.

We can see that only the RASFF data source returns data instances that are labelled as CED, Expiration date, HC, or Illegal Importation. Additionally, only the EMA data source is used to extract data instances classified as theft and resale. This observation suggests that the Data Source feature has the most influence on the model's forecast. Product Category and Origin Country are also regarded as features that primarily influence the model's forecast.

Observations were also made while XAI tools were being implemented. Some of them are in line with earlier findings reported in (Kuzlu et al., 2020) about LIME and SHAP. A total of 100 data instances were used in the Shapley values calculation. The computation of Shapley values takes longer the more data there is. As a result, not all of the data instances could be used. The same data instances and models are always used by SHAP to produce the same results. In contrast to SHAP, LIME's results, which were generated at diverse periods, varied from one another. This problem is caused by an algorithm that messes with randomly chosen data instances. Both SHAP and LIME generate graphs that offer regional or global justifications. WIT, on the other hand, lacks this capability; instead, it offers an interface that enables users to assess the behaviour of models, view counterfactuals, and view the data distribution of the dataset. WIT produced some results for us, however, it calls for manual work. It is fairly simple to use, and this tool does not require the use of any specific model. Users can quickly explore a dataset using WIT.

Data from RASSF and EMA, two independent databases, are retrieved to create the dataset. If the dataset includes more data points from other databases, the XAI tools' output can be different. We got findings using two different XAI tool types. Other tools may yield comparable outcomes. The dataset used in this study had missing values, hence some imputation techniques were used to fix this issue. There are some more imputation methods, though, that we have not yet researched. The performance of the prediction model and the outcomes of the XAI tools may both be impacted by changing the imputation methods. The dataset's total amount of data points was constrained. The performance of the deep neural model and the output of XAI tools are both influenced by the amount of training data used. Another discovery is that various dataset characteristics, such as Fraud complexity and Demand rise, have distinct characteristics that were shown by WIT prior to model construction. These characteristics do not add anything to the model.

The model presented in this research can be applied to different datasets. Fine-tuning of the model may be needed, but the main steps of the approach will not be affected. The quality and scale of the new dataset can affect the model's performance. As the new dataset might have different characteristics compared to the one used in this research, the results can be slightly different.

7 | CONCLUSION AND FUTURE WORK

In order to comprehend the thinking behind the DL model decisions and provide an explanation for a particular prediction, this research attempted to develop a DL model to predict the type of food fraud. It also applied the explainable AI tools SHAP, LIME and WIT. The authors of this article created the dataset used in this research in a previous investigation. The findings show that explainable AI tools can provide information to interpret the model and results as well as insight into the behaviour of DL models. The tools used to apply the model are evaluated in a number of areas, including execution speed, explanation style (globally vs. locally), and usability. When selecting an explainable AI tool for this area, these dimensions can offer some guidance.

This study's main finding is that the data source feature (EMA and RASSF) has the greatest impact on the model's prediction. It seems that not all instances of food fraud are reported equally in these databases. Despite the small sample size, this analysis provides valuable information about anticipating food theft. We believe that these findings can alleviate hesitation to use AI models and can inspire specialists who operate in the food supply chain to apply DNN models.

There are several implications of this study for practitioners and researchers. We believe that XAI research can help bridge the gap between ML models and human decision-making in the food safety domain and stakeholders can be involved more effectively (Ryo, 2022). Also, consumers, regulators and other stakeholders can better understand how food safety risks are predicted and managed with the help of ML models (Borrego-Díaz & Galán-Páez, 2022). For researchers, more accurate models can be developed by analysing and explaining the prediction of the existing models because the strengths and weaknesses of the model can be observed with the help of XAI models (Kawakura et al., 2022; Wells & Bednarz, 2021).

The following issues are considered for future work:

1. Adding new data instances from different databases could improve the present DL model's performance and provide more insight into the observed trends.
2. The XAI tools used in this study have a few shortcomings that need to be addressed in future research.
3. More precise XAI tools can be developed and used to provide more information about models without requiring user intervention.

ACKNOWLEDGEMENTS

This research was partly supported by the Knowledge Base (KB) programme (KB-38 Data driven High Tech (DDHT), which is financed by the Dutch Ministry of Agriculture, Nature and Food Quality (LNV). Open Access funding provided by the Qatar National Library (QNL)

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are available from Wageningen Food Safety Research Institute. Restrictions apply to the availability of these data, which were used under license for this study. Data are available from the author(s) with the permission of Wageningen Food Safety Research Institute.

ORCID

Okan Buyuktepe  <https://orcid.org/0000-0002-1366-5111>

REFERENCES

- Aldughayfiq, B., Ashfaq, F., Jhanjhi, N. Z., & Humayun, M. (2023). Explainable AI for retinoblastoma diagnosis: Interpreting deep learning models with LIME and SHAP. *Diagnostics*, 13(11), 1932.

- Arrieta, A. B., Díaz-Rodríguez, N., del Ser, J., Bennetot, A., Tabik, S., Barbado, A., & Herrera, F. (2020). Explainable artificial intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Information Fusion*, 58, 82–115.
- Bießmann, F., Rukat, T., Schmidt, P., Naidu, P., Schelter, S., Taptunov, A., Lange, D., & Salinas, D. (2019). DataWig: Missing value imputation for tables. Accessed 10 January 2022. <http://jmlr.org/papers/v20/18-753.html>.
- Borrego-Díaz, J., & Galán-Páez, J. (2022). Explainable artificial intelligence in data science: From foundational issues towards socio-technical considerations. *Minds and Machines*, 32(3), 485–531.
- Bouzembrak, Y., & Marvin, H. J. (2015). Prediction of food fraud type using data from rapid alert system for food and feed (RASFF) and Bayesian network modeling. *Food Control*, 61, 180–187.
- Castelvecchi, D. (2016). Can we open the black box of AI? *Nature*, 538, 20–23. <https://doi.org/10.1038/538020a>
- Christoph, M. (2020). Interpretable machine learning. Lulu.com. Accessed 10 January 2022. <https://christophm.github.io/interpretable-ml-book>.
- Dieber, J., & Kirrane, S. (2022). A novel model usability evaluation framework (MUsE) for explainable artificial intelligence. *Information Fusion*, 81, 143–153.
- Fukuda, K. (2015). Food safety in a globalized world. *Bulletin of the World Health Organization*, 93, 212. <https://doi.org/10.2471/BLT.15.154831>
- Getting Started. (2022). Getting Started. Accessed 10 January 2022. https://docs.seldon.io/projects/alibi/en/latest/overview/getting_started.html.
- Goodman, B., & Flaxman, S. (2017). European Union regulations on algorithmic decision-making and a “right to explanation”. *AI Magazine*, 38(3), 50–57.
- Käferstein, F. K., Motarjem, Y., & Bettcher, D. W. (1997). Foodborne disease control: A transnational challenge. *Emerging Infectious Diseases*, 3, 503–510.
- Kaur, H., Nori, H., Jenkins, S., Caruana, R., Wallach, H., & Wortman Vaughan, J. (2020). Interpreting interpretability: Understanding data scientists' use of interpretability tools for machine learning. In *Proceedings of the 2020 CHI conference on human factors in computing systems* (pp. 1–14). ACM.
- Kawakura, S., Hirafuji, M., Ninomiya, S., & Shibasaki, R. (2022). Analyses of diverse agricultural worker data with explainable artificial intelligence: XAI based on Shap, LIME, and LightGBM. *European Journal of Agriculture and Food Sciences*, 4(6), 11–19.
- Kuzlu, M., Cali, U., Sharma, V., & Guler, O. (2020). Gaining insight into solar photovoltaic power generation forecasting utilizing explainable artificial intelligence tools. *IEEE Access*, 8, 187814–187823. <https://doi.org/10.1109/ACCESS.2020.3031477>
- Le Cun, Y., Bengio, Y., & Hinton, G. E. (2015). Deep learning. *Nature*, 521, 436–444.
- Lombardi, A., Diacono, D., Amoroso, N., Monaco, A., Tavares, J. M. R., Bellotti, R., & Tangaro, S. (2021). Explainable deep learning for personalized age prediction with brain morphology. *Frontiers in Neuroscience*, 15, 578.
- Lundberg, S. M., & Lee, S. I. (2017). A unified approach to interpreting model predictions. In *Proceedings of the 31st international conference on neural information processing systems* (pp. 4768–4777). Neural Information Processing Systems Foundation, Inc. (NeurIPS).
- Mutukumira, A. N., & Jukes, D. J. (2003). The development of national food safety control systems in sub-Saharan Africa – Issues and opportunities. In *Proceedings of Food Africa conference, 5–9 May 2003, Congress Centre/Palaisde Congress, Yaoundé, Cameroon*.
- Nogales, A., Morón, R. D., & García, A. (2020). Food safety risk prediction with deep learning models using categorical embeddings on European Union data.
- Parsa, A. B., Movahedi, A., Taghipour, H., Derrible, S., & Mohammadian, A. (2020). Toward safer highways, application of XGBoost and SHAP for real-time accident detection and feature analysis. *Accident Analysis & Prevention*, 136, 105405. <https://doi.org/10.1016/j.aap.2019.105405>
- Raschka, S. (2018). MLxtend: Providing machine learning and data science utilities and extensions to Python's scientific computing stack. *Journal of Open Source Software*, 3(24), 638.
- Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). Why should i trust you?: Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 1135–1144). ACM.
- Ryo, M. (2022). Explainable artificial intelligence and interpretable machine learning for agricultural data analysis. *Artificial Intelligence in Agriculture*, 6, 257–265.
- Sahay, S., Omare, N., & Shukla, K. K. (2021). An approach to identify captioning keywords in an image using LIME. In *International conference on computing, communication, and intelligent systems (ICCCIS)* (pp. 648–651). IEEE. <https://doi.org/10.1109/ICCCIS51004.2021.9397159>
- ScikitLearn Documentation StandardScaler. Accessed 10 January 2022. <https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html>.
- Shapley, L. S. (1953). A value for n-person games. In *Contributions to the theory of games* (Vol. 2, pp. 307–317). Princeton University Press.
- Shrikumar, A., Greenside, P., & Kundaje, A. (2017). Learning important features through propagating activation differences. In *International conference on machine learning* (pp. 3145–3153). PMLR.
- Skater. (2022). Skater overview. Accessed 10 January 2022. <https://oracle.github.io/Skater/overview.html#skater>.
- TeamHG-Memex. (2022). TeamHG-Memex/eli5. Accessed 10 January 2022. <https://github.com/TeamHG-Memex/eli5>.
- Wells, L., & Bednarz, T. (2021). Explainable AI and reinforcement learning—A systematic review of current approaches and trends. *Frontiers in Artificial Intelligence*, 4, 550030.
- Wenbo, G., Dongliang, M., Jun, X., Purui, S., Gang, W., & Xinyu, X. (2018). LEMNA: Explaining deep learning based security applications. In *Proceedings of the 2018 ACM SIGSAC conference on computer and communications security (CCS '18)* (pp. 364–379). Association for Computing Machinery. <https://doi.org/10.1145/3243734.3243792>
- Wexler, J., Pushkarna, M., Bolukbasi, T., Wattenberg, M., Vi'egas, F., & Wilson, J. (2019). The what-if tool: Interactive probing of machine learning models. *IEEE Transactions on Visualization and Computer Graphics*, 26(1), 56–65.

AUTHOR BIOGRAPHIES

Okan Büyüktepe is a senior software engineer with a strong educational background. He received a bachelor's degree in electrical and electronics engineering from Gaziantep University, Turkey, in 2016. Furthermore, he pursued his higher education at Bahçeşehir University in Turkey, where he completed his Master's degree in Computer Science. During his master's studies, he deepened and expanded his expertise in computer science. He focused on areas such as software development, database management, and data analytics. Throughout this period, he actively engaged in challenging software projects and played an integral role in their execution, utilizing advanced programming languages

and tools. He developed a particular interest in artificial intelligence, machine learning, and big data analytics. By exploring new technologies and trends in these fields, he aimed to continually update and broaden his knowledge. Currently, he is working as a senior software engineer, contributing to various projects in the telecommunications industry. His areas of specialization include application development and data analysis.

Cagatay Catal is a Full Professor at Qatar University, Department of Computer Science & Engineering. He obtained his BSc and MSc degrees in Computer Engineering from Istanbul Technical University in 2002 and 2004, respectively. He later pursued his Ph.D. in Computer Engineering at Yildiz Technical University in Istanbul, completing it in 2008. From 2020 to 2021, he held the position of Full Professor in the Department of Computer Engineering at Bahcesehir University in Istanbul. Prior to that, he spent two years as a full-time faculty member at Wageningen University & Research (WUR) in the Netherlands. Before his time at WUR, Dr. Catal worked for six years in the Department of Computer Engineering at Istanbul Kultur University, where he served as an Associate Professor and Head of the Department. In April 2014, he was awarded the title of Associate Professor by the Inter-University Council of Turkey. Before entering the academic field, Dr. Catal had 8 years of experience at the Scientific and Technological Research Council of Turkey (TUBITAK), Information Technologies Institute, where he held the positions of Senior Researcher, Project Manager, and Senior Software Engineer. His research interests encompass various areas, including Artificial Intelligence, Deep Learning, Machine Learning, Natural Language Processing, Software Engineering, Software Testing, Software Architecture, and Precision Agriculture.

Gorkem Kar received his Ph.D. in Electrical and Computer Engineering from Rutgers University in 2017, where his research focused on mobile computing, location privacy, and connected vehicles. His research interests include vehicular networks, image processing, deep learning, object detection and classification, machine learning, and blockchain. Dr. Kar has published several papers in top academic conferences and journals, including Applied Soft Computing, MobiCom, MobiSys, and CCS. He is also a member of IEEE and ACM.

Yamine Bouzembrak is an Assistant Professor in Artificial Intelligence with a background in computer science and automation. He obtained his Ph.D. in computer science and automation from the Université Lille Nord de France in 2011, specializing in the field. Since joining Wageningen University and Research in 2014, Dr. Bouzembrak has focused his expertise on developing cutting-edge machine learning models that connect the realms of food, human health, and environmental sustainability. His work revolves around harnessing the power of AI to create models capable of effectively monitoring, detecting, and preventing potential risks within the food system, all while considering the broader impact.

Hans Marvin (MSc, PhD) studied chemistry/ biochemistry and has worked on various topics related to biotechnology and food science. In 1991, he joined Wageningen University & Research (WUR) and became senior scientist in Wageningen Food Safety Research (WFSR), previously RIKILT. In this organization, he worked on many aspects of food safety. The main topics of research were emerging risk & early warning applying Big Data and Artificial Intelligence (AI). On these topics he has organized and chaired numerous workshops and is author and co-author of 120 peer-reviewed scientific publications. Currently, Dr. Marvin is a senior scientist at HAYAN GROUP B.V., which applies Artificial Intelligence (AI) and Big Data technologies to offer quality high end services to their clients.

Anand K. Gavai is a researcher who focuses on agriculture and food systems. He specializes in utilizing data to gain valuable insights and is particularly skilled in privacy-preserving data platforms. In this field, he designs and implements data infrastructures and develops customized machine learning algorithms. These algorithms effectively integrate data from diverse sources by employing semantic web technologies. Driven by a deep commitment to sustainable change, Dr. Gavai aims to optimize farming practices and enhance resource management. His extensive background in both computer science and agricultural sciences allows him to construct robust frameworks for efficient data collection and analysis. By harnessing the power of AI, these algorithms derive actionable insights from a wide array of data, including weather patterns, soil composition, crop yields, and market trends. With his expertise in semantic web technologies, Dr. Gavai promotes collaboration and facilitates knowledge sharing within the agricultural community. He is dedicated to making a positive impact on agriculture and food systems. Currently, his work supports the development of a more efficient, resilient, open, and sustainable global food supply chain."

How to cite this article: Buyuktepe, O., Catal, C., Kar, G., Bouzembrak, Y., Marvin, H., & Gavai, A. (2023). Food fraud detection using explainable artificial intelligence. *Expert Systems*, e13387. <https://doi.org/10.1111/exsy.13387>

APPENDIX A

Figures A1–A7 represent the ROC and PR curves at different categorization levels.

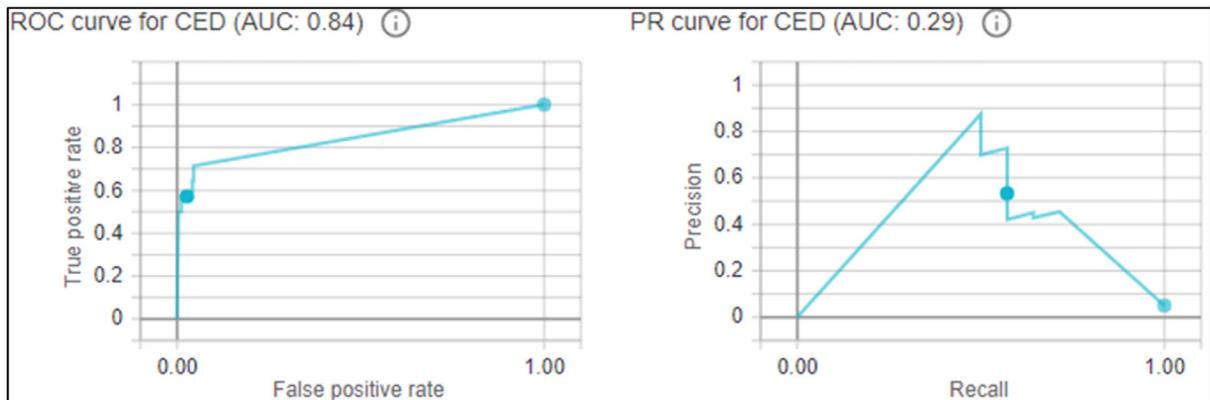


FIGURE A1 Receiver operating characteristic (ROC) and precision-recall (PR) curves for common entry document (CED).

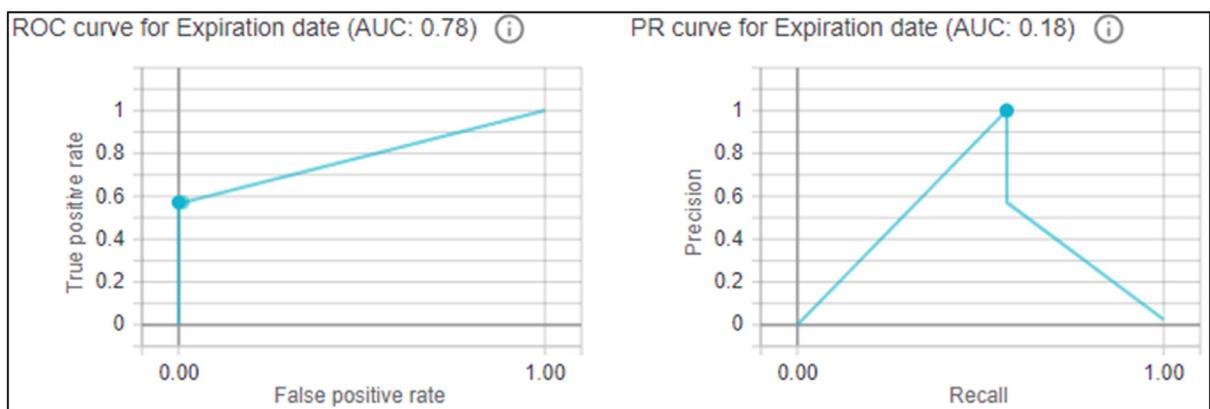


FIGURE A2 Receiver operating characteristic (ROC) and precision-recall (PR) curves for the expiration date.

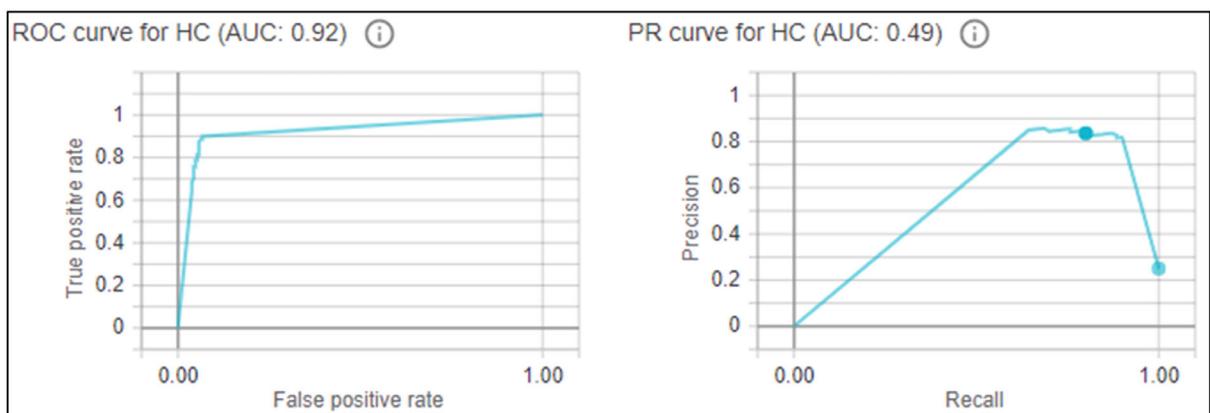


FIGURE A3 Receiver operating characteristic (ROC) and precision-recall (PR) curves for health certificate (HC).

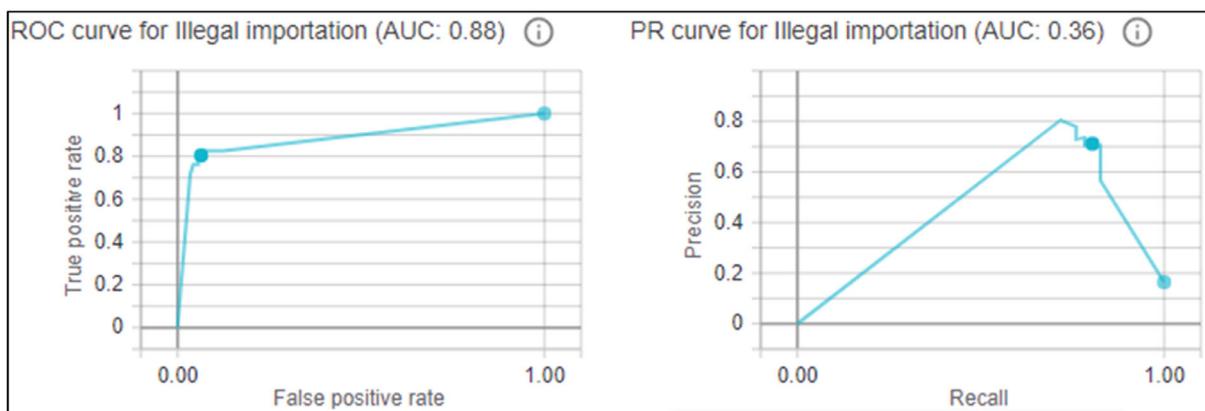


FIGURE A4 Receiver operating characteristic (ROC) and precision-recall (PR) curves for illegal importation.

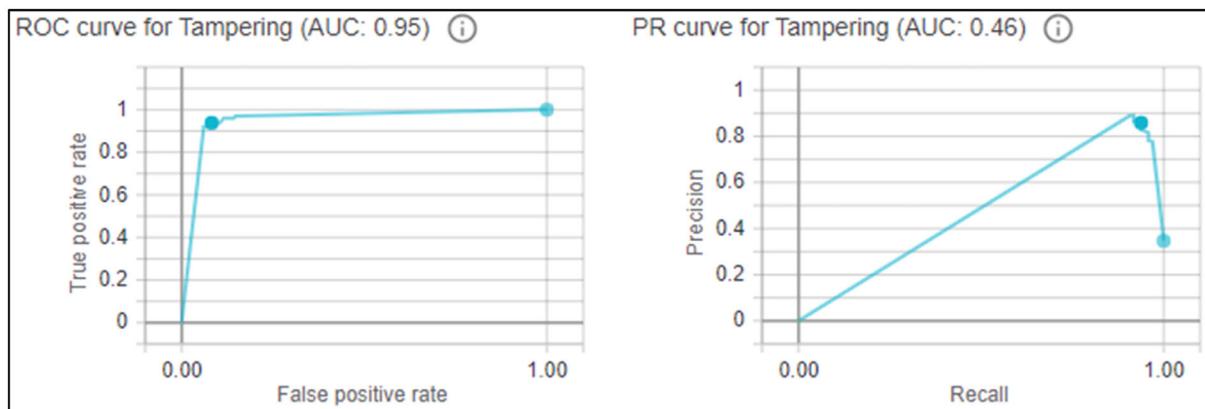


FIGURE A5 Receiver operating characteristic (ROC) and precision-recall (PR) curves for tampering.

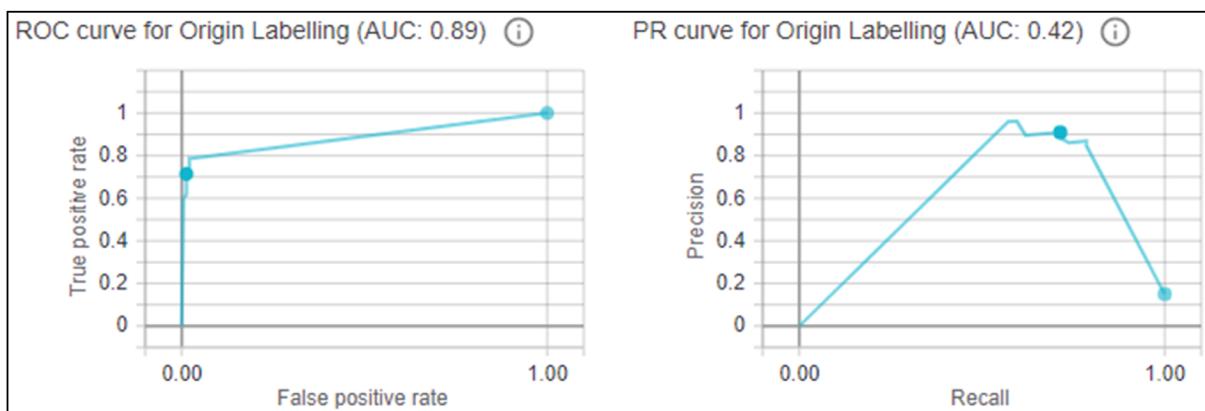


FIGURE A6 Receiver operating characteristic (ROC) and precision-recall (PR) curves for origin labelling.

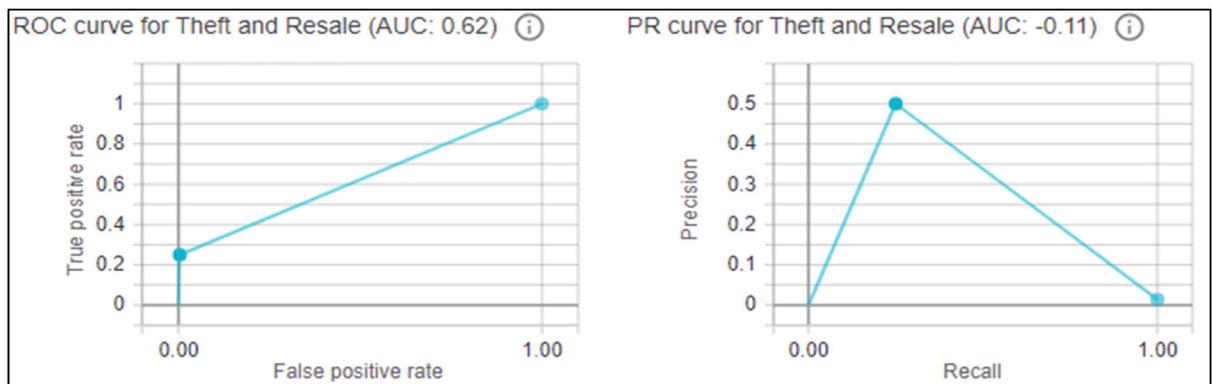


FIGURE A7 Receiver operating characteristic (ROC) and precision-recall (PR) curves for theft and resale.