A comparison of neural and non-neural machine learning models for food safety risk prediction with European Union RASFF data

Abstract. European Union launched the RASFF portal in 1977 to ensure cross-border monitoring and a quick reaction when public health risks are detected in the food chain. There are not enough resources available to guarantee a comprehensive inspection policy, but RASFF data has enormous potential as a preventive tool. However, there are few studies of food and feed risk issues prediction and none with RASFF data. Although deep learning models are good prediction systems, it must be confirmed whether in this field they behave better than other machine learning techniques. The importance of categorical variables encoding as input for numerical models should be specially studied. Results in this paper show that deep learning with entity embedding is the best combination, with accuracies of 86.81%, 82.31%, and 88.94% in each of the three stages of the simplified RASFF process in which the tests were carried out. However, the random forest models with one hot encoding offer only slightly worse results, so it seems that in the quality of the results the coding has more weight than the prediction technique. Our work also demonstrates that the use of probabilistic predictions (an advantage of neural models) can also be used to optimize the number of inspections that can be carried out.

Keywords: Food and Feed Safety, Machine Learning, Deep Learning, Random Forest, Entity embedding, Prediction.

1 Introduction

Friedman (2007) defines globalization as the "inexorable integration of markets, transportation systems, and communication systems to a degree never witnessed before". It increasingly involves more people and more economic sectors every day. The food and feed sector is perhaps one of the most affected, as the safety of food production and transport is a very sensitive issue for the consumer. The World Trade Organization[1] (WTO) estimates that food and agricultural products account for approximately 10% of all exports, making it a major concern for the transport industry and food safety authorities, as demonstrated by the economic impact and health risk of recent food crises. Ashworth and Mainland (1995) shows the drop in beef consumption after the outbreak of mad cow disease. Tuffs (2011) points to the need for accountability, as demonstrated by the disastrous failure to detect the origin of E. coli-contaminated cucumbers.

[1] https://data.wto.org/

In response to the global nature of food hazards, the independent European Food Safety Authority (EFSA) was established in 2002. EFSA enforces the key political priority of the European Commission to establish high standards of food safety, Jukes & Jukes and Mutukumira (2003). Together with the EU Commission, the EU Member States, Iceland, Liechtenstein, and Norway are part of RASFF[2], the Rapid Alert System for Food and Feed. RASFF provides authorities with an effective network to exchange information and react quickly and effectively when a health threat occurs. The network consists of contact points working at the national level who are responsible for registering all information in an online system called the RASFF Portal[3].

Health risk alerts are detected through market, food business, or border controls at each RASFF country. Each Member State is operational on a 24/7 basis, so urgent notifications can be made at any time. Food safety authorities organize their inspection activities considering available human and financial resources. As a non-exhaustive explanation of the RASFF process reveals, a single notification involves many people, resources, and efforts. Food inspectors select suspected products to be inspected. Depending on the selected product, different tests and analyses may be carried out, each specific to a different hazard. If the product is irregular, the competent authority decides how to deal with the problem and whether it should be reported to RASFF using an online form with details of the incident.

However, the number of inspections carried out is very small compared to the volume of the food and feed trade. It is estimated that in some cases only about 11% of the imported products can be analyzed.[4] An accurate prediction of which products are most likely to pose a risk, or which contaminant is most likely to be found at any given time, can help optimize the resources available for inspections and speed up detection, as they can focus on carrying out more specific analyses.

Prediction refers to the outcome of a model fitted to a training dataset when applied to unseen data. The model is an estimator that can make predictions of the future behavior of new events, basically a classification of future events among several, well determined, possibilities. A particular case of prediction is forecasting, where the model predicts a future numerical value based on the analysis of time-series data. Prediction is an inherently difficult task that, to be reliable, also requires a large amount of data, this being an additional problem to the prediction itself.

The rise in recent years of artificial intelligence and especially Machine Learning (ML) offers a possibility to address this problem from a different approach. ML techniques are mathematical models capable of identifying data patterns after being trained on a training dataset and using them to make decisions automatically when applied to fresh data. They have demonstrated their good performance in extracting patterns when the amount of data is so large that many details cannot be perceived by the human eye. A special type of ML model is the Artificial Neural Networks (ANNs), which are a simplified simulation of how certain areas of the human brain function. Among the

[2] https://ec.europa.eu/food/safety/rasff_en

[3] https://webgate.ec.europa.eu/rasff-window/portal/

[4] https://ec.europa.eu/food/sites/food/files/safety/docs/oc_leg_imports_dpe_ms_border-checks-results_2013.pdf

ANNs, Deep Learning models (DL) have been a major breakthrough in recent years, offering good results in prediction problems. Deep learning is performed by deep neural networks, a subset of ANNs. They are defined as models composed of multiple processing layers that learn representations of data with multiple levels of abstraction, LeCun et al (2015).

Despite the relevance of predicting food safety risks, Chlebicz and Śliżewska (2018), Moura et al. (2019), and Battilani et al. (2016) show that the problem of prediction in food and feed safety has not yet been studied in depth. Few works approach the problem within the RASFF framework or use the information stored in its portal for this purpose and none of them fully exploit the huge amount of data stored there. In addition, there are no studies on which is the best predictive strategy or which of the available techniques, whether the more classical ML or the more novel neural approaches, is more suitable for this problem.

This paper presents how the use of different ML techniques can help solve the problem of prediction in the field of food and feed safety and thus increase the chances of targeting specific products and/or contaminants that present a higher risk at a given time. In particular, we have made a comparison between neural and non-neural ML models that have been used to predict three different issue characteristics within a simplified RASFF workflow. First, the product category that will cause the issue so that authorities can focus on analyzing these products. Then, the hazard that will cause the problem so that resources can be allocated to the most appropriate analysis. Finally, what action can be taken to address this problem. Data availability is not a problem as we make use of the complete dataset stored in the RASFF portal since 1979. Data is downloaded automatically and periodically from the RASFF portal and pre-processed using data mining techniques to feed predictive models with properly formatted data.

Our research predicts incoming alert notifications using two neural models (multilayer perceptrons and 1D convolutional neural networks) and five non-neural ML techniques (logistic regression, decision trees, random forest, boosting trees, and support vector machines). Most machine learning algorithms (neural and non-neural) cannot work with categorical data and require all inputs and outputs to be numerical. For this reason, the categorical variables in the dataset used had to be encoded using different strategies (integer encoding, ~~binary encoding~~, feature hashing, one-hot encoding, and entity embedding) that have also been evaluated in combination with the predictive models.

The results show that there is a strong dependence on both the ML model and the coding strategy. Neural models with entity embedding perform best in terms of accuracy, although not all non-neural models perform in the same way. Decision trees also provide good results in combination with the one-hot encoding strategy.

This paper is organized in the following sections: Section 2 briefly explains the main contributions of related works in food safety predictions and contextualizes our approach. Section 3 describes the RASFF data collection process and the pre-processing before use as input to the models. Section 4 describes the methods and models used in this study, including encoding techniques and neural architectures. Section 5 shows the

experimental results, including the performance of the neural models used and the comparison with the results obtained with non-neural techniques. Finally, Section 6 presents the conclusions obtained in this research study and future work.

2 Background

Prediction in the field of food safety has been done using both classical ML and DL techniques. Fuzzy cognitive maps are used by Birmpa et al (2015) to detect critical points in a production food chain. Wang et al. (2018) use a bayesian tool for the prediction of foodborne diseases. Deep learning is used by Zuo et al. (2017) to predict safety in meat products. Ganguli et al. (2012) use a CNN trained with satellite pictures to predict key food security metrics while Ashqar et al. (2018) use images to detect diseases in tomato leaves. The particular use of DL techniques in different applications related to food is compiled in Zhou et al. (2019). In particular, it should be highlighted Song et al. (2017) which apply autoencoders to predict morbidity in gastrointestinal infectious diseases by using different types of food and their contaminants. These works, however, do not rely on the information contained in the RASFF portal nor do they consider the steps followed in the RASFF protocols, as we do.

RASFF data has already been used in several non-prediction works. Nepusz et al. (2009) use Network Analysis for analysis of food alert patterns from 2003 to 2008 concluding that China, Iran, and Turkey are the countries with the most problems. Petroczi et al. (2010) also use network analysis with food notifications from 2000 to 2009, the main finding reports that countries with important ports are a good method to preserve food safety. The rest of the papers in this paragraph are quantitative analyses using statistical methods. Luth et al. (2019) make a statistical analysis of issues with Listeria monocytogenes that recommends better communication channels between food safety and public health authorities. D.'Amico et al. (2018) work only with data related to seafood products from 2011 to 2015, in most of the cases products were imported from Spain and Italy finding heavy metals and pathogenic microorganisms. Taylor et al. (2013) measure how European Union Member States contribute to the RASFF, it concludes that there is a wide range of food policies comparing the different countries. Pigłowski (2019) analyzes only food issues related to pathogenic and non-pathogenic microorganisms, results recommend cooperation with RASFF to improve public health law that will reduce outbreaks related to microorganisms. Stanciu (2019) uses RASFF issues with Romanian products, concluding that authorities should worry due to the large number of exports withdrawn by the European Union. Kleter et al. (2009) work only for four years from 2003 to 2007 with the whole RASFF whose results suggest using complementary information as safety assessments, risk management measures, background hazards, or surveillance patterns. In Alshannaq and Yu (2021), products issued with mycotoxins in the US from 2010 to 2019 are analyzed, it recommends the implementation of a mandatory and enforceable legal framework to avoid nut trees being contaminated by aflatoxins. Taghouti et al. (2015) analyze data from the Mediterranean Partner Countries to study if previous notifications affect the actual ones. A study of contaminated dairy products from the last 20 years is made in Postolache et al.

(2020) where the results conclude that cheese products with microbial contaminants are the biggest hazard. The analysis in Leuschner et al. (2013) work with data concerning histamine concentrations establishes a link between this hazard and adverse effects on human health. Finally, Kowalska and Manning (2020), measure the importance of interpreting RASFF data to take decisions, demonstrating the EU members act in different ways when reporting issues.

Although these works make use of RASFF data, none of them make predictions or use ML techniques nor do they use the whole dataset provided by the RASFF portal. The only reference found so far related to our work is Bouzembrak and Marvin (2015), using RASFF data to predict a particular feature of health warnings applying Bayesian networks. However, our work presents a comparison between neural and non-neural techniques. We report an efficient DL approach to predict a set of features in different cases and create a training dataset using all the historical data in the RASFF.

3 Dataset

The predictive capacity of an ML model is highly dependent on the quality and quantity of training data. Therefore, the first step for this research is the acquisition and adequate pre-processing of raw data to obtain an appropriate training dataset.

3.1 Data source

Raw data were downloaded from the RASFF database, which has been in use since 1979. At the moment of the research, a total of 56,385 records were stored, each one corresponding to an alert notified by a Member State. Since the RASFF portal only allows downloading of the latest 5,000 records, in a human-readable format which is not suitable for digital processing, we used web scraping techniques that generate structured data based on the unstructured data available on the website, Saurkar et al. (2018).

We develop a web scraper in a Python script that uses different libraries such as Selenium[5], to download information; Time[6], to avoid a timeout that causes the interruption of the script and CSV module[7], to read and write CSV files. The output of the script is a CSV file that contains all the historical records of a certain time interval that can be updated at any time. Each row corresponds to a registered alert and each column to a feature of that alert. All the features of the 56,385 alerts were scraped but only the characteristics relevant a priori to this study were stored. All except date and subject correspond to categorical variables written in English. Table 1 contains a brief description of each feature and its possible values.

[5] https://selenium-python.readthedocs.io/index.html
[6] https://docs.python.org/2/library/time.html
[7] https://docs.python.org/3/library/csv.html

Table 1. Description and values for the features downloaded from RASFF records.

Once the dataset was downloaded and stored, minor pre-processing was required to ensure the format integrity and correctness and to avoid errors when encoding the variables. Several Python scripts removed duplicates or characters that could cause an error, while others replaced empty strings and NaN values with a blank value. The last script format and renamed categories are written differently when, for example, the value has been written with or without capital letters or the name has changed over the four decades of RASFF's operation. Classical Python libraries like pandas[8] or NumPy[9]were used for this pre-processing.

3.2 Training Dataset

Once we have a cleaned and formatted dataset, the next step is to select the period over which we are going to train the models. This is done by analyzing the distribution of data over the years. Scraped data has been plotted chronologically with Matplotlib[10]. As can be seen in Fig. 1, 50,416 records out of 56,385 occurred between 2004 and 2019 at a rate of about 3,000 food records per year. This represents 88.93% of the total so that before 2004 the number of issues was low (about 100 records per year in the period [1979-1997] and 500 in the period [1998-2003]). This difference of incidents per year can lead to biases in the predictions and these residual records can cause more inconveniences than benefits when training the models. For this reason, the final dataset will only contain 50,416 records from 2004 to 2019.

Fig. 1. Records per year registered in RASFF.

However, due to the way ML/DL models are trained, they can overfit. The model overfits if it maximizes the model's performance on the training set and performs poorly on the unseen data. The model then begins to memorize the training data rather than learning to generalize from it. A common practice to avoid this behavior is to split the dataset into three subsets and keep one part as a "validation set", on which the training performance is measured, and another as a "test set", on which a final evaluation is performed. This double division aims to prevent the model to overfit to the validation set if the hyperparameters are excessively modified in search of optimal performance. In this case, the evaluation metrics would not represent the quality of the generalization. Training is therefore carried out on the training set, after which an evaluation is performed on the validation set. When the experiment appears to be successful, a final evaluation is done on the test set.

The experiments conducted for this work used data from 2004 to 2018 as the training and validation set (with a proportion of 80-20% of records) while we have preserved 2019 data for testing. This decision was based on the time series and annual seasonality of food and feed alerts. The testing instances were not seen by the models during the

[8] https://pandas.pydata.org/
[9] https://numpy.org/
[10] https://matplotlib.org/

training stage and are entirely new to them. By dividing the available data into three sets, the number of samples that can be used is drastically reduced and the results may depend on a particular random choice for the training/validation sets. To avoid this problem k-fold cross-validation has been used. The training set is divided into k smaller sets or "folds". Then a loop is carried out for each of the k folds: the model is trained using k-1 folds as training data and validated (measured its performance) with the remaining fold. The performance measure reported by the cross-validation is the average of the values calculated for the whole loop. For our experiments, we have used a value of k = 5, where each fold consists of 9,525 data. Table 2 shows the number of records in each dataset division.

Table 2. Number of records of each dataset used in the experiments.

4 Methods

4.1 Encoding of categorical variables.

Most of the machine learning algorithms cannot work with categorical data. They require all inputs and outputs to be numerical. For this reason, all variables in the dataset must be encoded, that is, converted into numbers that preserve as best as possible the information contained in the dataset. The choice of encoding type is directly related to the accuracy of the model. In this case, this impact can be even bigger as all the variables in the dataset are categorical. Therefore, the models proposed in this paper are the sum of the encoding and the machine learning models. The different types of proposed encodings, based on Hancock and Khoshgoftaar (2020), are presented below.

Integer encoding. Every category from a categorical variable is transformed into integers. It gives number 1 to the first category, 2 to the second, and so on, till *n* which is the number of different values that the categorical variable can take.

Feature hashing. This method creates a hash table using a function with the same name. It transforms input elements or strings of any length into output numerical code of a fixed length determined by a function.

One hot encoding. Each value of the categorical variable is represented by a vector of size n, where n is the number of possible values of the variable. In each vector, all positions will contain 0, except the one representing the coded category, which will contain 1.

Entity Embedding of Categorical Variables. Guo and Berkhahn (2016), propose this method that reduces the use of memory and accelerates the process of model formation compared to one-hot encoding.

For multi-dimensional spaces of categorical features, this method automatically maps closer categories with similar effects to the target output, so helping neural networks to solve the problem. In other words, entity embeddings are used to map categories into a continuously distributed vector in a semantic space. In this space, similar

categories are closer. What is even more interesting is that not only the distance between categories are meaningful but also the direction of the vectors. This allows finding the intrinsic properties of categorical variables.

This coding is done using embedding layers. Each categorical variable is an input to the model and needs a different inlay layer. That is, the relationships sought are the relationships that exist between the category values of each categorical variable, not the relationships that exist between variables. The output dimension of each of these layers is a hyperparameter that will be modified during the training process considering the expected output value of the training data. Fig. 2 shows how embedding layers are connected to the neural model.

Fig. 2. Example of a model with embedding layers.

The operation of an embedding layer can be understood as a table in which each different input (each category) is associated with an embedding vector of a pre-established dimension. Therefore, each category has a unique vector associated with it. The values of these vectors are randomly initialized and modified as if they were the weights of a neural network, depending on the final error made by the prediction. The representation of categories made by a model that solves a specific problem is expected to be one that retains only the features and relationships relevant to solving that specific problem.

The output vectors of each of the embedding layers are concatenated to each other representing the input data, being the union of the codifications of all the variables. This vector will, therefore, be fed into the model that follows the coding.

4.2 Neural models.

The experiments described in this paper apply deep neural network models to RASFF portal data to predict three different issues' characteristics, at different stages of a simplified RASFF workflow. Three different predictors have been developed, each one built with the required encodings and a particular neural architecture. Although several neural models have been tested, the best results have been obtained with Multilayer Perceptrons (MLP) and 1D Convolutional Networks (Conv1D).

Multilayer Perceptron (MLP) is a feedforward supervised neural network model. It consists of an input layer, an output layer, and an arbitrary number of hidden layers. The basic MLP has a single hidden layer. Neurons use nonlinear activation functions, either sigmoid, hyperbolic tangent, or Rectified Linear Unit (ReLU). Learning is carried out through backpropagation using the generalized delta rule to update the weights matrices.

1D Convolutional Neural Network (Conv1D) is a type of feedforward supervised deep neural network that can model high-level data abstractions using hierarchical architectures. They learn input-output relationships based on convolution operations over a one-dimensional array. Each convolutional layer extracts hierarchically and incrementally some characteristics input array.

Since the project deals with categorical variables, output layers for both MLP and Conv1D models are built with as many neurons as categories must be predicted. The activation function for these output layers is Softmax, which produces a value between 0 and 1 for each output neuron. That value is the probability that this neuron represents the correct output of the network. The sum of the outputs of all the neurons in the output layer is equal to 1. The loss function selected was categorical cross-entropy since the problem can be defined as a label-categorization task. All models have been developed using Keras[11], a high-level API built on top of TensorFlow[12], the Google open-source library for machine learning and deep neural networks.

Optimization of model hyperparameters. The behavior of deep neural networks is controlled by different hyperparameters, their selection being based on a trial and error basis. This is a time-consuming task, so a method called grid search has been being used to find an optimal combination. Grid search is a method that combines different values from a set of hyperparameters, obtaining the model with the best accuracy, Bergstra et al. (2011).

4.3 Non-neural models.

As part of the evaluation of the proposed models, some classical ML models have been used. They have also been used with the different proposed encodings except for categorical embeddings as they can only be used with neural models. These models have been developed with the help of Python library scikit-learn[13] which is used for Data Mining.

Logistic regression. It is a technique that belongs to what is called linear generalized models. The main characteristic of this model is that it can predict a qualitative variable based on several predictive ones. According to Park (2013), it analyzes the relationship between multiple independent variables and a categorical dependent variable.

Decision trees. This model consists of predicting the target variable by learning simple decision rules inferred from the data features. It is defined in Mashat et al. (2012) as a classifier that uses the values of the attributes to create a recursive partition based on the input data. It builds logic diagrams in the form of hierarchical trees. They represent the categorization of the data under a series of conditions applied in the form of successive trees. In the tree, each node represents a test or decision on an attribute, and each branch the result of the test, and each node finishes in a class label.

Random forest. Introduced by Breiman (2011), it is a method based on decision trees. The technique consists of combining them and averaging the models improving the results. Compared with decision trees, this technique usually reduces overfitting which is a problem in them.

Boosting trees. This technique is also based on decision trees and is defined in Anghel et al. (2018) as an algorithm that uses a set of decision trees and builds a learner by

[11] https://keras.io/

[12] https://www.tensorflow.org/

[13] https://scikit-learn.org/stable/

fitting the gradients of the residuals. It makes use of the different techniques that optimize and improve gradient descent over the loss function.

Support Vector Machine (SVM). The actual version of SVM was proposed by Boser et al. (1992). It can be seen as a classifier where instances are distributed through an n-dimensional space. The objective of the algorithm is to find a hyperplane that divides individuals into different classes making the separation between them as wide as possible. In this case, we have performed two versions of SVM for classification and another for regression (SVR).

4.4 Food and feed prediction system.

To maximize the results of this work concerning the use of different ML techniques for food safety risk prediction, we have designed a simplified RASFF model. This model captures the main steps and actors involved in the workflow and decision-making process. Using this model and following extensive research, we concluded that RASFF can take advantage of this work in three key points, defined as the three stages of the workflow.

- Prediction of product category at the time officials at the border or in a market decide which products might be contaminated. A series of products having different characteristics arrive from different countries. Due to the limited amount of human resources, not all the products can be analyzed. By predicting which products have more possibilities of being contaminated, they can focus only on taking samples of a few products.
- Prediction of hazard category in the laboratory, once the product to be analyzed has been selected. A single product can present multiple hazards, each one requiring a different analysis approach. At this point, there is an interest in predicting the hazard that can be found in the product, so the usage of the laboratory equipment is optimized.
- Action to be taken once both the product and the hazard are detected. This is a decision-making process

As the use cases are sequential, the output of one stage will be part of the inputs for the next one. Table 1, lists input features for each stage, and the range of different values that the predicted variable can take. Input data of Stage 1 are, a priori, known by the EU workers when the goods arrive at a country.

Table 3. Inputs and outputs of the models at different stages.

5 Results

Selection of the neural model. Two models (a multilayer perceptron and a 1D convolutional neural network) have been developed and trained with a combination of classical encodings and entity embedding of categorical variables.

Neural architectures have been designed and tuned using iterative grid search techniques applied to the two possible models (MLP and Conv1D) with one hot encoding

(standard coding). A first iteration was used to select the number of neurons, hidden layers, and the activation function for each layer. From these results, the dropout between different hidden layers was fixed. In a new iteration, the number of epochs and optimizers were selected. The last hyperparameter that was set was the optimal learning rate. For the Conv1D model, the same process was followed except in the first iteration, where the number of neurons is changed for the number of filters, kernel sizes, and Maxpooling. Tables 4 and 5 show the hyperparameters that have been tested and their different values for the MLP and the Conv1D, respectively. The final configuration of the three models can be seen in Appendix Fig. 1, 2, and 3.

Table 4. Configuration of grid search for MLP (hyperparameters and values).

Table 5. Configuration of grid search for Conv1D (hyperparameters and values).

Tables 6, 7, and 8 show the accuracy obtained on the validation set at each stage with both deep learning models when input data was coded with one hot encoding (the best results among classical encodings) and entity embeddings of categorical variables. The experiments have been carried out five times each, starting each time with randomly generated weight matrices. The results are the average and standard deviation of the five runs thus avoiding deviations from the mean that could be obtained with a single experiment. The accuracy is measured in the output layer of each model as the number of times the most likely category matches the correct one divided by the total number of input values.

Best accuracies in stages 1 and 3 are obtained with an MLP with entity embedding of categorical values. For stage 2, the best result is achieved with the same encoding but using a Conv1D.

Table 6. Accuracy of the different neural models. Stage 1.

Table 7. Accuracy of the different neural models. Stage 2.

Table 8. Accuracy of the different neural models. Stage 3.

The stage 1 model consists of four embedding layers of size 6, 16, 9, and 50 that are concatenated with an MLP with three hidden layers of 2048, 1024, and 512 neurons. The activation function for these layers is ReLU which represents a nearly linear function and therefore preserves the properties of linear models that made them easy to optimize, with the gradient descent method, Goodfellow et al. (2018). The output layer has 38 neurons that correspond to the product categories. Error is measured with the categorical cross-entropy and Adam has been used as the optimizer.

The stage 2 model is made up of one more embedding layer of 19 neurons. These layers are concatenated with a Conv1D of two-layer of 512 and 256 neurons. The number of the convolutional filters is 128 and 256 with sizes 4 and 3 respectively. The output layer has 35 neurons (number of hazard categories).

Finally, the model at stage 3 is similar to model 1 with two differences: two extra embedding layers with 19 and 18 neurons and an output layer composed of 24 neurons that correspond to the different actions to be taken.

Evaluation against non-neural models. The proposed neural architectures have performed quite well. However, they need to be validated against classical ML models of non-neural nature that we have previously defined: logistic regression, decision trees, random forest, boosted trees, SVM, and SVR. These architectures alongside the different proposed encodings have been combined creating new models plus grid search technique to be compared with the chosen neural architectures from the previous subsection. Results of accuracy using validation set can be seen in Tables 9, 10, and 11, again one table for each stage. In the case of SVR, we have used R^2 as evaluation metric which is depecited in the following Equation.

$$R^2 = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \bar{y}_i)^2} \quad (1)$$

Table 9. Accuracy comparison of neural and non-neural models with different encoding strategies in stage 1.

From all the combinations of non-neural architectures, decision trees with one-hot encoding are the one that performs best. It obtains an accuracy of 81.93%. MLP with categorical embedding performs better with 86.81% of accuracy.

Table 10. Accuracy comparison of neural and non-neural models with different encoding strategies in stage 2.

Regarding Table 10, the best non-neural model is the combination of decision trees with one hot encoding with around 80.89% of accuracy. The neural architecture performs slightly better with an 82.31%

Table 11. Accuracy comparison of neural and non-neural models with different encoding strategies in stage 3.

At stage 3, the best non-neural results are provided again by decision trees and one hot encoding (around 81% of accuracy). In this case, the neural architecture works much better, improving the results by 7.04%.

Probabilistic predictions. Analyzing the results from the previous tables, we can conclude that the differences between neural and non-neural models are not significant. What is good about the neural models is that they provide probabilistic predictions. In this case, the output in all three neural models is a vector whose size is the number of neurons in the output layer. The value in each position varies between 0.0 and 1.0, adding a total of 1.0. This is interpreted as the probability that this position occurs expressed in times one.

Non-neural models make a single bet on what the expected category will be. Neural models with a probabilistic outcome can be used to narrow the scope of products, hazards, or actions to a small number of options (not just one). This broadens the scope of

the search but still guarantees a higher success rate. Following this approach, three different predictions have been made: Top1, Top2, and Top3.

- Top1. It compares the category to which the network gives the greatest probability with the real one. It is the accuracy used to measure the models in Tables 7, 8, and 9.
- Top2. In this case, the accuracy is calculated by checking whether the actual category is between the 2 most probable categories according to the neural model and adding both accuracies.
- Top3. The same as Top2 but checking if the actual category is among the 3 categories to which the network has given the most probabilities and adding the accuracies in the three categories.

Table 12 summarize the results obtained at each stage with the same models but calculating accuracies as described above. Top3 models logically improve accuracy.

Table 12. The accuracies of the three models depending on the type of prediction.

Looking at the results from the models in Table 12, the probabilistic predictions taking into account Top2 and Top3 categories obtain much better results compared with the non-neural models.

Evaluation of the prediction. Finally, all selected non-neural and neural models have been evaluated against the test dataset, previously separated from training/validation data. For that purpose, all 2,789 instances from the 2019 dataset, that have never been fed into the model, have been used. In order to measure how well the models are performing, we have also obtained other metrics like specificity and sensitivity. Specificity measures the ratio between the number of true negatives and the total of those predicted as true negatives and false positives. Sensitivity is the same as specificity but takes into account false negatives instead of false positives. Both metrics are depicted in the following Equations.

$$Specificity = \frac{number\ of\ true\ negatives}{number\ of\ true\ negatives + number\ of\ false\ positives} \quad (2)$$

$$Sensitivity = \frac{number\ of\ true\ positives}{number\ of\ true\ positives + number\ of\ false\ positives} \quad (3)$$

Table 13 compiles the information related to specificity and sensitivity for the best non-neural models in Tables 9 to 11. These results have been obtained with the test set.

Table 13. Different test metrics for the best non-neural model.

In the case of neural models, the prediction has been carried out for the three stages and accuracy has been calculated regarding probabilistic predictions: Top1, Top2, and Top3. The chosen architectures correspond to the best performances in Tables 6 to 8. This kind of prediction will improve the results and is particular to the neural models, thus these results can not be obtained with non-neural techniques. All this information

is summarized in Table 14. In the case of Top1, apart from accuracy, we have provided sensitivity and specificity calculated as described before.

Table 14. Different test metrics for each neural model.

6 Conclusions and future works

The experiments described in this document provide a detailed comparison of the prediction of food and feed risk issues obtained using different machine learning models. These models are applied at three specific stages, each of them yielding an intermediate result in the workflow of a simplified RASFF system. When all models are sequentially applied in a pipeline, risk product, hazard to be found in the product and preventive action can be predicted. The data has been obtained from RASFF using a scrapper specifically developed to obtain all the information stored in the RASFF portal. Models are both neural and non-neural machine learning methods combined with different coding strategies for categorical variables in the dataset.

The most suitable neural model for each stage of the workflow has been selected from two possible architectures (Multilayer Perceptrons and Convolutional Neural Networks). After a thorough hyperparameter tuning process, the selected MLP and 1DConv architectures have been trained and validated in all three stages. In two cases (stages 1 and 3) neural model stands out clearly over the other. In the worse case, the differences are about 1.42% (stage 2).

A first and straightforward conclusion is that the encoding strategy plays a prominent role in the quality of the results. More elaborate encoding strategies yield better results, regardless of the model. In all stages, the best results for all non-neural models are obtained when using one-hot encoding. For neural models, the best results are obtained with the entity embedding of categorical variables, followed by one-hot encoding. Entity embedding cannot be used with the other models, since the output dimension of each embedded layer is modified during the training process of the neural model to which it is associated, in much the same way as the weight matrix is optimized. The use of entity embeddings made a clear advantage of the neural model over non-neural ones.

Comparing neural and non-neural models at the validation stage, each with the best encoding strategy, the study shows that the use of entity embeddings plus neural models gives better results in all cases. The accuracy results of the neural models versus the average accuracy of the non-neural models clearly shows a clear advantage in favor of the MLP and 1DConv architectures. In addition, they have also been tested on data never used in the training/validation phase (2,789 instances of the 2019 dataset). A comparison of the results depicted in Tables 13 and 14 show that the neural models are robust, as the accuracies obtained are quite similar, without a large deviation in the test trials compared to the results obtained in the validation phase.

However, if we compare with the accuracy at the validation stage of the best non-neural model, which has been random forest and decision trees depending on the stages,

the advantage of the MLP/1DConv architectures are significant in some cases, performing much better than non-neural models with One Hot Encoding.
This scenario improves the results when considering, for neural models, not a categorical prediction but a probabilistic prediction. As the output of the neural architectures is a vector with as many values as categories must be predicted at each stage, the accuracy of each model has been validated in three different scenarios: a prediction based on the most likely category (Top1 prediction) and two predictions where accuracy is measured on the two (or three) most likely categories. Their performance is improved when the second and the third most probable prediction (Top2 and Top 3 predictions) are also considered. This is a fundamental advantage since these probabilistic predictions (which cannot be obtained with a non-neural model) have enormous potential as a preventive tool to optimize resources for the actors involved in food security. To check the accuracy of the models, we can take into account the specificity and sensitivity. From these results, we can conclude that accuracy values make sense because specificity obtains greater results. This is because this metric takes into account all the false positives. So, the models are good at minimizing the prediction of contaminated products that are really healthy for consumers. This is very interesting as the aim of this research is to optimize the resources.

As future works, some ideas can be highlighted. First, the work could benefit from using the seasonal nature of the products in the RASFF alerts. This means, forecasting food and feed problems through a time series approach. For example, we can make a weekly or monthly count of issues by type of food. From this, it is possible to predict when and how many instances of each type will occur at a given time. This will reflect the importance of the seasonality and, also, the hemisphere from where the products are imported. Second, we could use the variable summary of each record. This variable consists of a text written by the different authorities. The objective would be to obtain helpful information using Natural Language Processing (NLP) techniques. Third, we can understand RASFF as a big network where countries are connected to each other taking into account the products they are trading. So, based on graph theory techniques like Social Network Analysis (SNA), we can make a structural analysis of this network. We can find sensitive routes, countries that have food policies impacting the European Union, or hubs of countries whose border controls are finding a lot of contaminated products. Finally, as we can model the different registered issues as a graph, we can take advantage of the newly Graph Convolutional Neural Networks (GCNN) to train a predictive model of sensitive routes.

Acknowledgments

We would like to thank Enrique Ruiz Ruiz, Javier Caracuel Cayuela, and Fernando Pérez Lara for helping in the development of some experiments.

References

1. Alshannaq, A., & Yu, J. H. (2021). Analysis of EU Rapid Alert System (RASFF) Notifications for Aflatoxins in Exported US Food and Feed Products for 2010–2019. *Toxins*, *13*(2), 90
2. Anghel, A., Papandreou, N., Parnell, T.P., Palma, A.D., & Pozidis, H. (2018). Benchmarking and Optimization of Gradient Boosted Decision Tree Algorithms. *CoRR, abs/1809.04559*.
3. Ashqar, B. A., & Abu-Naser, S. S. (2018). Image-based tomato leaves diseases detection using deep learning.
4. Ashworth, S. W., & Mainland, D. D. (1995). The Economic Impact of BSE on the UK Beef Industry. Outlook on Agriculture, 24(3), 151–154. https://doi.org/10.1177/003072709502400304
5. Battilani, P., Toscano, P., Fels-Klerx, H.J., Moretti, A., Leggieri, M.C., Brera, C., Rortais, A., Goumperis, T., & Robinson, T. (2016). Aflatoxin B1 contamination in maize in Europe increases due to climate change. Scientific reports.
6. Bergstra, J. S., Bardenet, R., Bengio, Y., & Kégl, B. (2011). Algorithms for hyper-parameter optimization. In Advances in neural information processing systems (pp. 2546-2554).
7. Birmpa, A., Vantarakis, A., Anninou, A., Bellou, M., Kokkinos, P., & Groumpos, P. (2015). A user-friendly theoretical mathematical model for the prediction of food safety in a food production chain. *Journal of Food Processing & Technology*, *6*(2), 1.
8. Boser, B. E., Guyon, I. & Vapnik, V. (1992). A Training Algorithm for Optimal Margin Classifiers. *Computational Learning Theory* (p./pp. 144-152).
9. 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, vol. 61, pp. 180-187.
10. Breiman, L. (2001). Random forests. Machine learning, 45(1), pp.5–32.
11. Chlebicz, A., & Śliżewska, K. (2018). Campylobacteriosis, Salmonellosis, Yersiniosis, and Listeriosis as Zoonotic Foodborne Diseases: A Review. International journal of environmental research and public health.
12. D.'Amico, P., Nucera, D., Guardone, L., Mariotti, M., Nuvoloni, R., & Armani, A. (2018) Seafood products notifications in the EU Rapid Alert System for Food and Feed (RASFF) database: Data analysis during the period 2011–2015. Food Control, 93, 241–250
13. Friedman, T. L. (2007). The World is Flat: The Globalized World in the Twenty-first Century. Penguin. ISBN: 0141034890
14. Ganguli, S., Dunnmon, J., & Hau, D. (2019). Predicting food security outcomes using convolutional neural networks (cnns) for satellite tasking. *arXiv preprint arXiv:1902.05433*.
15. Goodfellow, I., Bengio, Y., Courville, A., & Bengio, Y. (2016). *Deep learning* (Vol. 1, No. 2). Cambridge: MIT press.
16. Guo, C. & Berkhahn, F. (2016). Entity Embeddings of Categorical Variables. *CoRR*, abs/1604.06737.
17. Hancock, J. T., & Khoshgoftaar, T. M. (2020). Survey on categorical data for neural networks. *Journal of Big Data*, *7*(1), 1-41.
18. Jukes D. J. & Mutukumira A. N. The development of national food safety control systems in Sub-Saharan Africa – issues and opportunities. In Proceedings Food Africa Conference, pp 5-9. Cameroon. (2003)
19. Kleter, G.A., Prandini, A., Filippi, L., & Marvin, H.J. (2009). Identification of potentially emerging food safety issues by analysis of reports published by the European Community's Rapid Alert System for Food and Feed (RASFF) during a four-year period. Food and

chemical toxicology: an international journal published for the British Industrial Biological Research Association, 47 5, 932-50.

20. Kowalska, A., & Manning, L. (2020). Using the rapid alert system for food and feed: potential benefits and problems on data interpretation. *Critical reviews in food science and nutrition*, 1-14.
21. LeCun, Y., Bengio, Y. & Hinton, G. E. (2015). Deep learning. Nature, 521, 436-444.
22. Leuschner, R. G., Hristova, A., Robinson, T., & Hugas, M. (2013). The Rapid Alert System for Food and Feed (RASFF) database in support of risk analysis of biogenic amines in food. *Journal of food composition and analysis*, *29*(1), 37-42.
23. Luth, S.S., Boone, I., Kleta, S. & Al Dahouk, S. (2019). Analysis of RASFF notifications on food products contaminated with Listeria monocytogenes reveals options for improvement in the rapid alert system for food and feed, Food Control, 96, pp. 479-487.
24. Mashat, A.F., Arabia, S., Fouad, M.M., Yu, P.S., & Gharib, T.F. (2012). A Decision Tree Classification Model for University Admission System. *International Journal of Advanced Computer Science and Applications (IJACSA)*, 3.
25. Moura, G.F., M Tomborelli, P., C T Carvalho, R., O Sigarini, C., T Carvalho, F., Vieira, B. & E S Figueiredo, E. (2019). Listeria monocytogenes and other species as persistent contaminants in the processing of chicken meat. The Journal of Applied Poultry Research.
26. Nepusz, T., Petróczi, A., & Naughton, D.P. (2009). Network Analytical Tool for Monitoring Global Food Safety Highlights China. PloS one.
27. Park, H.A. (2013). An introduction to logistic regression: from basic concepts to interpretation with particular attention to nursing domain. *Journal of Korean Academy of Nursing, 43 2*, 154-64.
28. Petroczi, A., Taylor, G., Nepusz, T. & Naughton, D. (2010). Gate keepers of EU food safety: Four states lead on notification patterns and effectiveness. Food and Chemical Toxicology, 48(7), 1957- 1964.
29. Pigłowski, M. (2019). Pathogenic and Non-Pathogenic Microorganisms in the Rapid Alert System for Food and Feed. International journal of environmental research and public health.
30. Postolache, A. N., Chelmu, S. S., Ariton, A. M., Ciorpac, M. I. T. I. C. Ă., Pop, C., Ciobanu, M. M., & Creangă, Ș. T. E. O. F. I. L. (2020). Analysis of RASFF notifications on contaminated dairy products from the last two decades: 2000-2020. *Romanian Biotechnological Letters*, *25*(2), 1396-1406.
31. Saurkar, A.V., Pathare, K.G., & Gode, S.A. (2018). An Overview On Web Scraping Techniques And Tools, International Journal on Future Revolution in Computer Science & Communication Engineering (IJFRSCE), Volume 4 Issue 4, PP: 363 – 367.
32. Song, Q., Zheng, Y. J., Xue, Y., Sheng, W. G., & Zhao, M. R. (2017). An evolutionary deep neural network for predicting morbidity of gastrointestinal infections by food contamination. *Neurocomputing*, *226*, 16-22.
33. Stanciu, S. (2019). 'Quality of Romanian Food Products on Rapid Alert System for Food and Feed Notifications'. World Academy of Science, Engineering and Technology, International Science Index 146, International Journal of Biological, Biomolecular, Agricultural, Food and Biotechnological Engineering, 13(2), 34 - 37.
34. Taghouti, I., Martinez Gomez, VD., García Alvarez-Coque, JM. (2015). Exploring EU food safety notifications on Agro-food imports: Are Mediterranean partner countries discriminated? International Journal of Food and Agricultural Economics. 3(2):15-29.
35. Taylor, G., Petróczi, A., Nepusz, T., & Naughton, D.P. (2013). The Procrustean bed of EU food safety notifications via the Rapid Alert System for Food and Feed: does one size fit

all? Food and chemical toxicology: an international journal published for the British Industrial Biological Research Association, 56, 411-8.
36. Tuffs, A. (2011). Cause of outbreak of E Coli in Germany is still uncertain. Brit. Med. J. 342: d3454
37. Wang, X., Zhou, M., Jia, J., Geng, Z., & Xiao, G. (2018). A Bayesian approach to real-time monitoring and forecasting of Chinese foodborne diseases. *International journal of environmental research and public health*, *15*(8), 1740.
38. Zhou, L., Zhang, C., Liu, F., Qiu, Z., & He, Y. (2019). Application of deep learning in food: a review. *Comprehensive reviews in food science and food safety*, *18*(6), 1793-1811.
39. Zuo, M., Cao, C., Zhang, Q., Wang, C., & Wang, H. (2017). Study on Quality and Safety Prediction of Meat Products Based on CNN. *DEStech Transactions on Computer Science and Engineering*, (iceiti).

Appendix

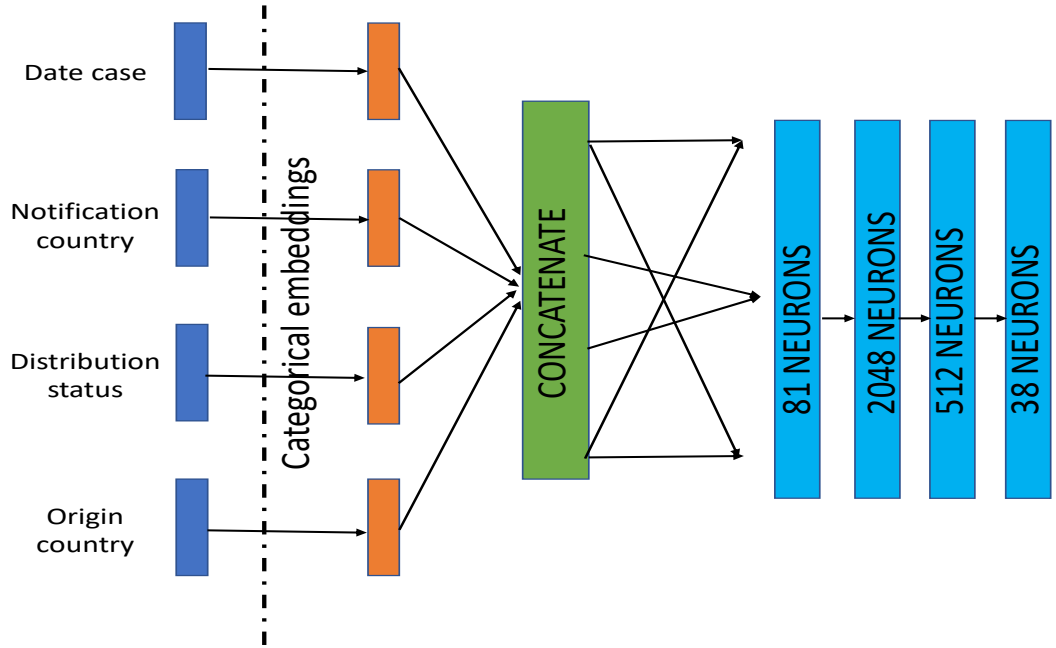

Fig. 1. Model for stage 1.

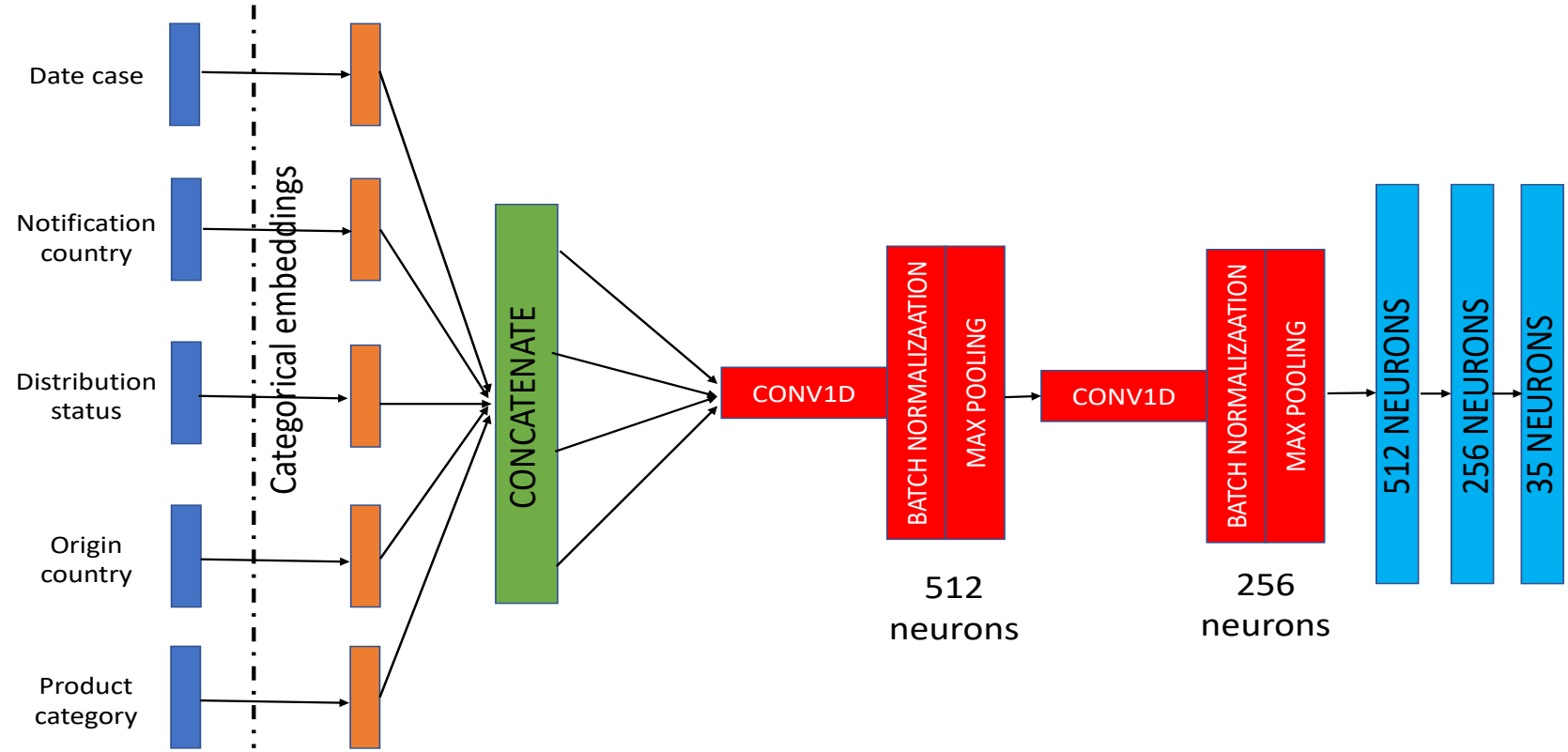

Fig. 2. Model for stage 2.

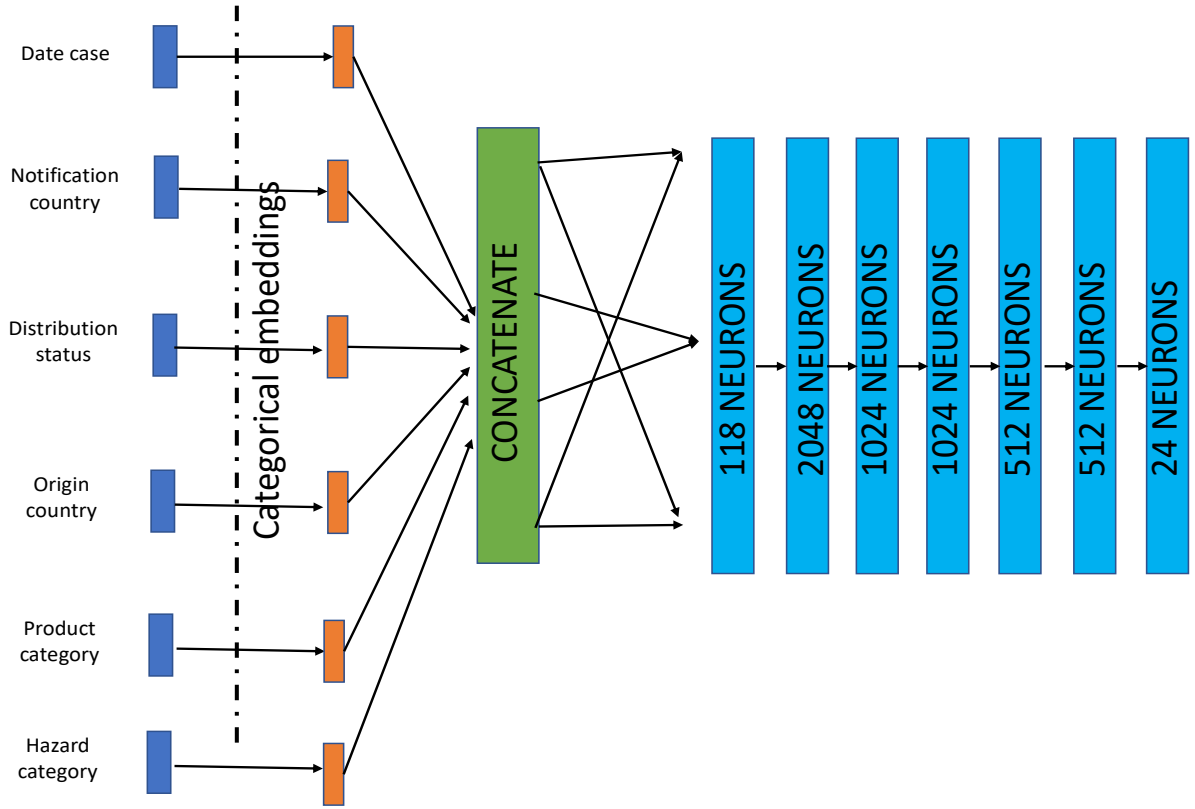

Fig. 3. Model for stage 3.