Toward Stable and Accurate Machine Learning Explanations: Optimizing LIME Stability Through Classifier Design

Joshua Prout*

Abstract

Machine learning and AI are being used for decision making in an increasing number of sectors, with models becoming deeper and more complex to capture intricate data patterns. The trade-off is that these complex models become less interpretable to human users, eventually becoming ”black boxes”, opaque to investigation by practitioners into how data features were used, and their contribution to the decision. Governments across the world, including the European Union, United States, and United Kingdom are demanding minimum levels of explainability when machine learning and AI are used in high risk sectors. Post-hoc explainability techniques such as LIME can provide estimated explanations for the prediction of individual data points from black box models, potentially fulfilling the explainability requirement while keeping the predictive power of the complex model.

Trust in LIME at its current state of development faces challenges from potential instability of explanations, when run multiple times on a single data point it can generate very different explanations each time. This is due in part to the random sampling involved in LIME's algorithm. This project reduces the instability by modifying the underlying black box model as part of a multi-objective optimisation. Multiple real world datasets for automated security threat detection were used for these experiments. It was found that LIME stability can be improved for ensemble models by limiting the depth of the model, with minimal impact on classifier performance. For imbalanced datasets, it was found that using SMOTE to address class imbalance had a positive impact on LIME stability.

I certify that all material in this dissertation which is not my own work has been identified.

Signature: _____

* JNP207@EXETER.AC.UK

Contents

Acknowledgments: I would like to extend my deep thanks to Dr David Walker for his invaluable guidance across this project, and the wider computer science faculty of the University of Exeter for their support

1. Introduction

A supervised machine learning model can be easily scrutinised with regards to its accuracy against the ground truth, satisfying us of its ability to predict and classify. Can we truly *trust* this model however, if it is so complex or deep that it is impossible for a human to ascertain how this decision was made? When a customer asks why a credit risk model has denied their loan, is it acceptable to tell them ”The machine can’t tell us”? Are key research insights being missed because models cannot explain the underlying patterns in data? Can machine learning be used in medical diagnostics without explanation of its decision?” These are examples where the idea of explainable Machine Learning shows its importance. The key focus of the research field has been to increase the ”explainability” of models, without reducing their accuracy. This is difficult due to the inherent tradeoff between model complexity, predictive power, and the interpretability of the model. Complex, non-interpretable models are often referred to as ”black boxes” for their opaque nature.

An example of a block box model dramatically impacting trust was seen when Apple introduced the Apple Card, which used machine learning to make decisions on credit limits. It was alleged that there was significant gender bias against women, receiving lower credit limits despite having a better credit history [1]. The company was eventually cleared of intentional discrimination, however they initially were not able to prove how the model made the limit decisions, and which features were used.

As machine learning and AI are rapidly introduced into decision making systems across society, lawmakers are beginning to see the need to address the black box model problem, and demand levels of explainability when machine learning and AI are used to make decisions about a person.

In March 2024, the European Parliament adopted the Artificial Intelligence Act. Article 86 of the new law states that any person subject to a decision made by an AI system *”shall have the right to obtain from the deployer clear and meaningful explanations of the role of the AI system in the decision-making procedure and the main elements of the decision taken”* [2].

As of April 2024, the Artificial Intelligence (Regulation) Bill is in the review stage in the Parliament of the United Kingdom. The act gives provision for an AI Authority, of which one of their roles is to ”deliver appropriate transparency and explainability” guidelines for AI practitioners [3].

With regard to explainability, machine learning models fall into two categories: the inherently explainable models, such as simple linear models or short decision trees. The weights and parameters of these models can be interpreted relatively easily to understand the impact and use of each feature. The coefficients of a linear model can be interpreted to give the contribution of each feature, a short decision tree can reveal the decision pathways down the branches. The downside of these models is that their simplicity often does not allow them to capture the complex nature and relationships of real-life data. Modern deep/complex models have achieved incredible accuracy on difficult datasets, at the expense of interpretability. Models such as deep neural networks, complex ensemble models including Random Forest and Extreme Gradient Boost are regarded as ”black box” models.

Research in explainable ML has developed techniques and methods that aim to increase the interpretability of a black box model after it has been trained, by providing explanations for individual predictions, or the general behaviour of the model. These are not always trusted due to the fact that they are they are only estimates of the underlying decisions of the model. A key area of research is increasing the accuracy and usefulness of these explanations.

This project will focus on a specific explanation method called LIME (Local Interpretable

Model Explanations). LIME relies heavily on random sampling and perturbation which can lead to instability in its output over multiple runs on the same data point. This project's aim is to reduce this instability through adjustment to the underlying black-box model structure and parameters, with the minimal possible impact on model accuracy, in the form of multi-objective optimisation experiments. This represents a novel area of research in a field that has previously focused solely on improvements to the LIME algorithm itself.

1.1. Overview of ML explainability techniques

Explainability techniques/methods can be split into a number of categories [4]. Firstly, there are intrinsic vs post hoc techniques. *Intrinsic* methods use the inner workings of the model itself, such as analysing weights of a linear model, or using the splits of a short decision tree. *Post-hoc* techniques are applied after a model has been trained and predictions have been generated. Methods can be split further into local or global explanations. *Local* methods provide an explanation for an individual data point, whereas *global* methods explain the general behaviour of the model. Further still, methods can be model specific or model agnostic. *Model agnostic* methods can be used on any model after training, whereas *model specific* methods, such as those that rely on interpretation of specific weights only work for one type of model, such as a decision tree.

One popular explainability method is SHAP [5], using an idea from game theory known as Shapley values. The method calculates a contribution for each feature by removing and adding features into subsets called coalitions, to determine the individual impact of each feature on the prediction of a data point. SHAP is a post-hoc, model agnostic, local method, however it has the option to give a global prediction by combining many individual feature importance calculations across the dataset.

Another popular method is the global surrogate model, an interpretable model trained to approximate a black box model across the full range of the data. This makes it post-hoc, model agnostic and has a global scope. Any interpretable model can be used depending on the task, however it is important to note that the surrogate relies firstly on the black box model that it is approximating being accurate, and the interpretable model may not approximate well enough to provide useful explanations [4].

There are many more explanation methods available, however this project will focus solely on LIME, a post-hoc, model agnostic, local scope model which will be explained in detail in the next section.

1.2. LIME

LIME [6], is a popular tool to explain the predictions of complex models. For a chosen data point it returns feature contributions, an estimate of which features had the most impact in making the specific prediction. It also shows whether the features had a positive or negative impact (pushing the model towards or away from the predicted class). LIME was chosen for this project as it has the potential to help comply with regulatory requirements by way of being a local scope model. By providing in depth explanations of how features of the data contributed to the final prediction, it has the potential to meet the requirements in article 86 of the EU safety bill: ”clear and meaningful explanations of the role of the AI system in the decision making procedure and the main elements of the decision taken” [2]. An example of a LIME explanation for a data point is shown in Figure 1, with the direction and magnitude of each feature's contribution towards the each classification category.

LIME works by fitting an interpretable model over the top of the decision space produced by the black box model. This model could be a linear model, a short decision tree or any other interpretable model. To fit this model, it first perturbs the dataset around the point to be explained, these new generated points are weighted by their distance to the point to be

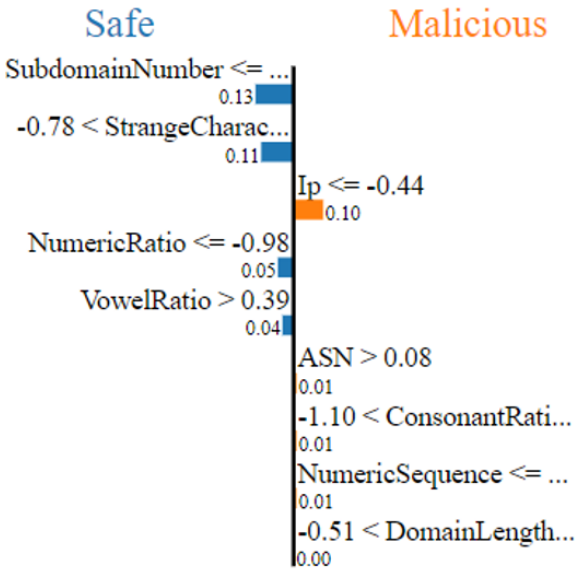

Fig. 1. Example LIME output

explained. These points are fed back into the black box model to get its prediction for each point. LIME then fits the interpretable model to these labelled points. The coefficients of the generated model are then used as an explanation as to whether the features had a positive or negative contribution to the class prediction of the selected point, and the magnitude of this contribution. The process on a simplified 2 dimensional dataset is shown in Figure 2. The scope of this project covers tabular data, however LIME can also be used on image and text data.

LIME represents a minimisation problem in the form [6]:

$$\xi(x) = \arg\min_{g \in G} \mathcal{L}(f, g, \Pi_x) + \Omega(g)$$

Where g is a model in the set G, defined as all models that are potentially interpretable to a user, such as a linear model or decision tree. $\Omega(g)$ represents the complexity of the model, such as the number of non-zero weights for a linear model, or the depth of a decision tree. f is the original, non-interpretable (black box) model, and Π_x is a proximity measure for weighting the generated samples around the point to be explained x. The function $\mathcal{L}$ measures how well g approximates f in the local space defined by Π_x. The model g that best approximates the black box model by minimising $\mathcal{L}$, and maintains high interpretability by minimising Ω is selected. This model is then interpreted to provide explanations of the behaviour of the underlying black box model for the selected point.

1.3. LIME instability and previous work

LIME has strong potential to increase the interpretability of complex models, especially with its model agnostic approach allowing it to be used with any model. A current drawback is the potential for instability of explanations produced, where over multiple runs of LIME for the same data point, wildly different explanations can be produced. This casts significant doubt onto whether the explanation faithfully represents the workings of the underlying black box model. Figure 3 shows two LIME runs on the exact same point and model, with instability in the importance and impact of each feature.

Zhang et al investigated the reasons behind the instability of LIME [7]. They found three key

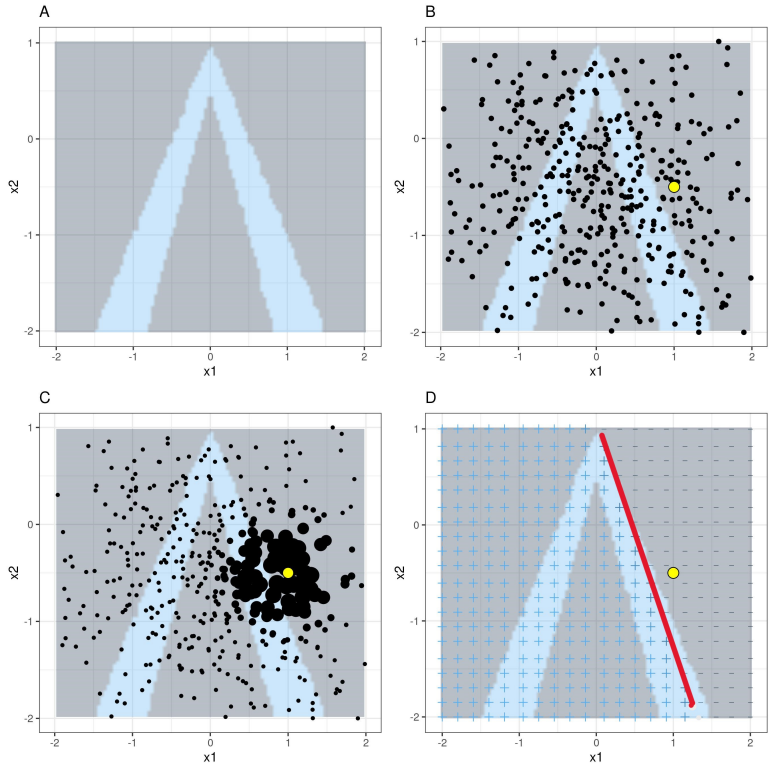

Fig. 2. 2 dimensional visualisation of the stages of fitting LIME to a dataset[4]. The background colours represent the decision boundaries of the underlying model. The black points represent the randomly perturbed points, their size representing distance weighting to the point to be explained in yellow. Finally, the red line represents the decision boundary of the interpretable model.

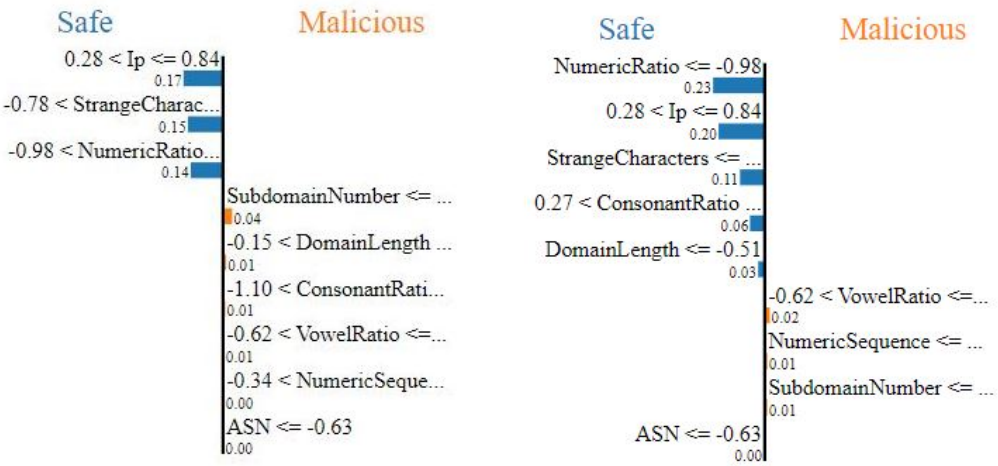

Fig. 3. Two unstable LIME runs on the same data point and model

factors that can cause and increase variance in LIME explanations. Firstly the random sampling procedure to generate points around the chosen point can cause different explanations for each run, depending on where the points fall on the local decision boundary. Secondly, variation can be exacerbated by the nature of the sampling, including the number of points generated and the calculation of proximity, to the point to be explained. Finally, variability can differ between points, certain points may have more stable explanations depending on the behaviour of the model around that point.

A number of approaches have been taken to increase the stability of LIME, each focused on the LIME algorithm itself. S-LIME [8] is an extension of LIME that seeks to address stabil-

ity by intelligently choosing the number of perturbations used to generate, using statistical hypothesis testing to justify the number chosen. Similarly, Opti-LIME [9] finds a balance between stability of the explanations and how closly they adhere to the underlying model. DLIME (Deterministic LIME) proposes removing the random sampling altogether, and using a deterministic version using Agglomerative Hierarchical Clustering and KNN to select data around the point to be explained [10].

Work so far in the LIME stability area have modified the LIME implementation, whilst leaving the underlying black-box model as a constant. This project takes a different approach, using the standard LIME implementation and modifying the underlying black-box model to enhance stability.

2. Project specification

This project investigates the relationship between the underlying machine learning model and the stability of LIME explanations on the models predictions. The scope of the project focuses on classification models, and tabular datasets. The core working principle is that the structure and parameters of the classification model change the shape and layout of the decision space (Figure 4) . LIME uses this decision space when fitting the interpretable model, by classifying the perturbed points according to the boundaries.

A more complex model will generally develop a more complex decision boundary, allowing for modeling of complex relationships and a more powerful predictive model. This project hypothesises that as the decision space becomes more complex, LIME instability increases, building on the findings from [7]. Success for this project will be finding optimal solutions such that LIME stability can be improved compared to the instability of the most accurate classifier, with as minimal impact on prediction accuracy as possible, giving rise to a multi-objective optimisation problem.

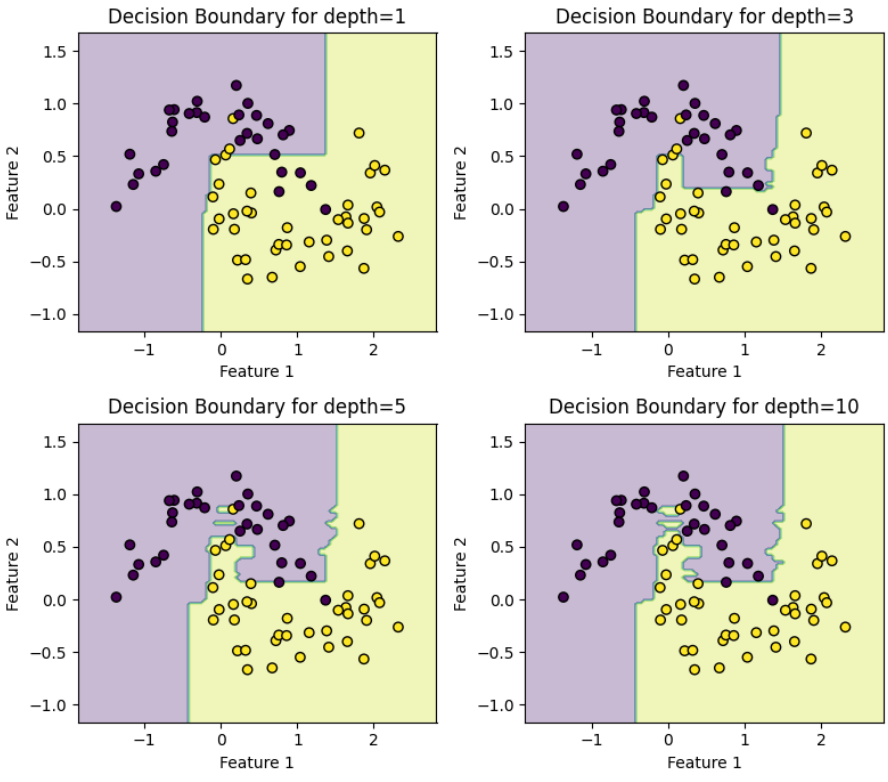

Fig. 4. 2D synthetic data visualisation of the impact of random forest depth on decision boundary complexity

2.1. Stability metrics

A vital part of this project is the quantitative definition of stability when evaluating LIME explanations. Over many repeated LIME calls for the same data point, an unstable LIME explainer will return very different contribution coefficients for each individual feature. Approaches to measuring this instability and quantifying it into one value for use in optimisation experiments are described in this section.

2.1.1. Visani LIME Coefficient Stability Index

A set of metrics developed by Visani et al can be used to evaluate the stability of LIME explanations [11]. They propose two metrics, the Variable Stability Index (VSI) and the Coefficient Stability Index (CSI). The VSI measures the stability of the subset of features selected by LIME to fit the interpretable model. If the subset is the same for each run of LIME on a datapoint, the VSI is 100.

The CSI measures the stability of the coefficient values by measuring the overlap of confidence boundaries for the coefficient values. The CSI is defined as such:

For any black box model f, set of P training features, data instance to be explained x, and subset of features to use in the explanation model p, LIME in its default setting fits a weighted ridge regression model g, giving a mapping between each feature and ridge coefficient R : $g : F \rightarrow R$.

m calls to LIME are made on the model f and the point to be explained x, giving a set of explainable models $g \dots g_m$. CSI then compares the coefficients of the models for each feature to test the stability, using algorithm 1:

Algorithm 1 Coefficients Stability Index (CSI) [11]

Input: $g_1 \dots g_m$
for feat in F **do**
    $M \leftarrow \{\}$
    for $i \leftarrow 1 \dots m$ **do**
        if g_i(feat) $\neq 0$ **then**
            $CI \leftarrow$ ConfInt(g_i(feat))
            $M \leftarrow M \cup CI$
        end if
    end for
    $n \leftarrow 0$
    for CIpair in $\binom{M}{2}$ **do**
        if OVERLAP($CIpair$) **then**
            $n \leftarrow n + 1$
        end if
    end for
    $PAR_{feat} \leftarrow \frac{n}{|\binom{M}{2}|}$
end for
$CSI \leftarrow$ mean(PAR)
Output: CSI

$$\text{ConfInt}(g(\text{feat})) = \left[g(\text{feat}) - 1.96\sqrt{\text{Var}(g(\text{feat}))}, \quad g(\text{feat}) + 1.96\sqrt{\text{Var}(g(\text{feat}))}\right]$$

$$\text{OVERLAP}(CI_{\text{pair}}) = \begin{cases} 0 & \text{if } CI_a \cup CI_b = \emptyset \\ 1 & \text{otherwise} \end{cases}$$

The outline of the algorithm relies on calculating confidence intervals for each coefficient value using the ConfInt function. The paper assumes a Gaussian distribution of the coefficients, and uses this to construct 95% intervals. For each unique pair of confidence intervals, the overlap is calculated to determine the CSI score, with a maximum score of 100 showing that all of the coefficients for each feature lie within the confidence boundary.

2.1.2. Weighted Coefficient of Variance metric

This project proposes a different approach for stability evaluation, based on the principle of coefficient of variance. The basic approach is the same, making multiple LIME calls, collecting the feature importance coefficient and then assessing the variance. The full algorithm is as follows:

Let $E = \{e_1, e_2, \ldots, e_n\}$ be a set of LIME explanations, where each explanation e_i is a set of tuples (f_{ij}, v_{ij}). Each tuple consists of a feature name f_{ij} and its corresponding coefficient v_{ij}. The feature coefficient is the contribution (positive or negative) that the feature had towards the prediction.

For each unique feature f, create a list L_f containing all values of f across all explanations in E.

For the top 5 most important features by magnitude of feature importance value, denoted as f_1, f_2, f_3, f_4, f_5, and their corresponding lists of values as $L_{f_1}, L_{f_2}, L_{f_3}, L_{f_4}, L_{f_5}$, calculate:

Mean μ_{f_i} and standard deviation σ_{f_i} for each L_{f_i}:

$$\mu_{f_i} = \frac{1}{|L_{f_i}|} \sum_{v \in L_{f_i}} |v|$$

$$\sigma_{f_i} = \sqrt{\frac{1}{|L_{f_i}|} \sum_{v \in L_{f_i}} (v - \mu_{f_i})^2}$$

Then, the coefficient of variance CV_{f_i} for each feature f_i is:

$$CV_{f_i} = \left(\frac{\sigma_{f_i}}{\mu_{f_i}} \right) \times 100$$

The weighted coefficient of variance for each feature f_i, considering its rank i (for the top 5 features), is given by:

$$CV'_{f_i} = \frac{CV_{f_i}}{i}$$

Weighting is performed due to the relative importance of the feature as you go down the importance list, variance in these features are less impactful to the overall trust of the explanation.

Calculate the average of the weighted coefficients of variation for the top 5 features as:

$$\overline{CV'} = \frac{1}{5}\sum_{i=1}^{5} CV'_{f_i}$$

2.1.3. Comparison of CSI and CoV

This project originally intended to use the CSI index [11] as the stability optimisation metric, however problems were encountered during the experiments. The first problem was the efficiency of the CSI Python implementation, the average runtime to calculate the stability of an explanation for one point was 17.5 seconds, wheras the CoV metric took on average 2 seconds. Optimisation experiments in this project involve calculating the stability for hundreds of data points for each model setting, therefore this processing time becomes greatly compounded

Secondly, results from CSI exhibited a high level of variance compared to CoV as seen in Figure 5. This is possibly due to the way CSI uses the overlapping confidence intervals, a very small change in coefficient value that causes the confidence interval to lose overlap causes a large change in the metric output. In CoV, a small change in coefficient values only leads to a small change in the metric output. The maximum value for the CSI score is 100, which it would often reach during experiments. This is not useful for this optimisation project, as we seek to optimise stability even when the explanation is suitably stable for normal use, to gain insights into the theory that can be applied to other, more unstable datasets.

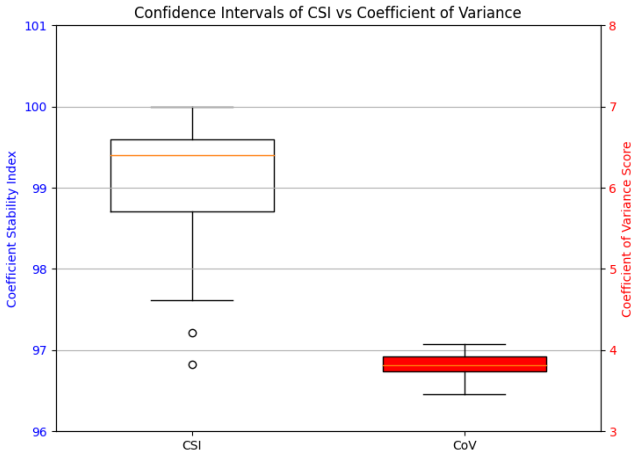

Fig. 5. Distribution of metric results for 50 runs on a single data point

2.2. LIME parameters

One of the most important LIME parameters is the number of perturbed samples used in model fitting. A balance needed to be found, as low sample size leads to high instability whereas a high sample size is very computationally expensive and leads to high runtimes, especially over thousands of LIME calls as required by the coefficient of variance algorithm. Figure 6 shows the relationship between size and stability, for these experiments 6000 points were chosen, however the optimal number will vary between datasets. The LIME algorithm

was kept as close to its default as possible, using the euclidean distance kernel and default ridge regression setting.

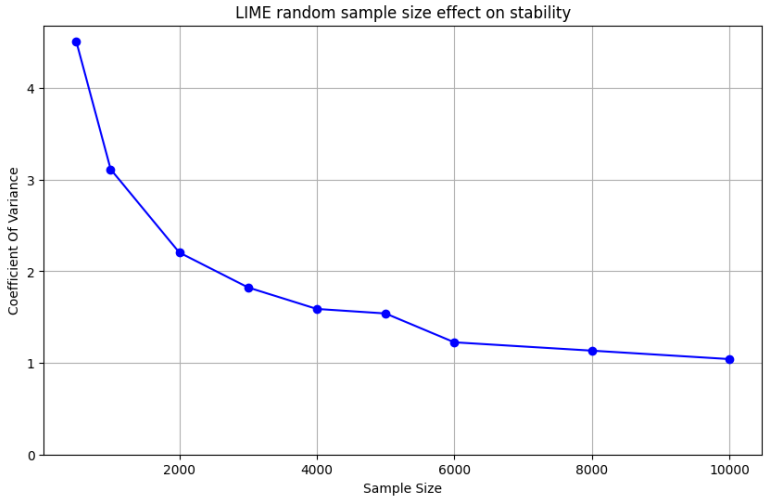

Fig. 6. Number of generated LIME samples against explanation stability

2.3. Classifier performance metric

To measure and compare classifier performance, the F1 metric will be used:

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

As the harmonic mean of precision and recall, the F1 score provides more useful information than simply an accuracy score, especially when dealing with imbalanced datasets such as the one used in this project.

3. Design

3.1. Datasets

This project focuses on classification datasets, using multiple real world examples in the field of automated security anomaly detection. Due to the high cost associated with mistakes and misinterpretations of data in areas such as web security and banking fraud, practitioners training models in this field are likely to benefit from reliable interpretability.

3.1.1. DNS dataset

The first dataset used for experiments in this project is a classification dataset for malicious and non-malicious web domains, based on logs from the Domain Name System [12]. Malicious domains are domain names registered or hijacked by hostile actors to carry out a range of harmful activities against users that unintentionally connect to the domain. Many automated approaches to detecting these domains exist [13] [14], this project will focus on using supervised machine learning to classify these domains, building on the existing work by Aslam et al [15]. Their paper trained a variety of models on this dataset, found their optimal parameters and then applied post-hoc explainability methods. Using these accuracy optimal models as a very useful starting point, this project will modify the structure and parameters of these models to optimise LIME explanation stability. The dataset has an advantage in

that it is well balanced 50/50 between the classes with 80,000 total samples. The paper [15] also determined the most useful features, using the Sequential Forward Feature Selection (SFFS) algorithm [16].

Feature	DataType	Description	Mean ($\pm$ Std dev)
TLD	Text	Top Level Domain	-
Ip	Text	Internet Protocol Address	-
ASN	Numeric	IP address associated with domain available	23,383.57 $\pm$ 30,049.9
DomainLength	Numeric	Domain name length	15.16 $\pm$ 3.539
SubdomainNumber	Numeric	Number of sub-domains in this DNS	1.007 $\pm$ 0.4248
VowelRatio	Numeric	Ratio of vowel characters	0.262 $\pm$ 0.099
ConsonantRatio	Numeric	Ratio of consonant characters	0.456 $\pm$ 0.146
StrangeCharacters	Numeric	Non-English language characters	4.35% $\pm$ 4.716
NumericSequence	Numeric	Maximum number of numerals	1.348 $\pm$ 1.555
NumericRatio	Numeric	Ratio of numeric characters	0.144 $\pm$ 0.147

TABLE I. Selected DNS dataset features with their data types, descriptions, and mean statistics.

3.1.2. Credit fraud dataset

The second dataset used is a popular dataset containing details of European credit/debit card transactions, classifying them into fraudulent or legitimate [17]. This dataset is massively imbalanced towards the negative class, with the positive class making up only 0.172% of all transactions in the dataset ($n = 284,000$). This gives an opportunity to study how the various approaches to tackling dataset imbalance impact LIME explanation stability, as well as investigating whether explanation stability differs between a minority and majority class.

The dataset has a relatively high dimensionality, with the majority of the features produced by Principle Component Analysis, and anonymised in public release for security reasons.

3.2. Models

A variety of models are used in this project:

Decision Tree: A generally inherently interpretable model when relatively shallow, decision trees are a good choice for explainable machine learning when explainability is the most important metric. Simple decision trees on their own will often struggle to effectively capture the complexity of some datasets, therefore decision trees will be used in this project as a benchmark for performance of a fully interpretable model compared to a black box model with post-hoc explainability techniques.

Random Forest [18]: This model trains many decision trees, each trained on a randomly generated boostrapped sample of the training dataset, creating a ”forest” of decision trees. When used to classify data, the final prediction is the mode of the predictions of all of the trees.

Extreme Gradient Boost (XGBoost) [19]: XGBoost builds on the principle of gradient boosted trees, whereby trees are build in sequence and fit on the errors of the previous tree. XGBoost increases the training speed by efficiently parallelising tree construction, and provides regularisation to prevent overfitting.

CatBoost [20]: Another decision tree boosting algorithm, CatBoost is designed to perform well with categorical data features, of which the DNS dataset described earlier has many.

Support Vector Classifier [21]: SVCs aim to find a decision plane that separates the classes with a maximum margin, using a kernel to raise data into higher dimensions. This project

whether the choice of kernel has an impact of LIME stability.

Neural Network: This project uses a relatively straightforward neural network, with 16 fully connected ReLu layers, then a dropout layer to reduce overfitting, then a sigmoid output node.

3.3. Software resources and experiment pipeline

Code for this project consists of a Python Jupyter notebook, to allow for demonstrability of the experiments and the flexible re-use of data and trained models. Jupyter further allows for the project to be hosted on the Google Colab platform, for powerful GPU acceleration if necessary. An important project management point was to not exceed the computation budget provided by the department. Key libraries for model training include SkLearn, TensorFlow, and the XGB and CatBoost open source libraries [22] [23].

4. Experiments and results

4.1. Optimisation on balanced dataset (DNS)

4.1.1. Ensemble models depth optimisation

The first experiment uses the DNS dataset with random forest, XGBoost and CatBoost models, investigating the impact of ensemble model depth on classifier performance and explanation stability. Figures 7, 8 and 9 show the relationship between these results and shows a correlation between F1 score and CoV instability. As the models become more complex and deep, their ability to classify the data increases, but so too does the LIME instability, keeping in mind that a higher CoV score indicates a more unstable explanation. These results are encouraging as it validates one of the assumptions of this project, that a more complex decision boundary causes higher LIME instability. This opens up the potential for multi-objective optimisation performed at the end of this section.

4.1.2. Support Vector Classifier kernel optimisation

When using a Support Vector Classifier, an important hyperparameter is the kernel used when training. This experiment investigated the relationship between kernel choice, classification performance and LIME stability (Figure 10). The simplest is the linear kernel, which produces slightly more stable LIME explanations but is less accurate at capturing the non-linear relationships in the data. The Radial Basis Function (RBF) has a higher F1 score and less misclassifications, with only a small increase in LIME variance. The sigmoid kernel produces the most stable explanations, however at the cost of a large F1 score drop. For clarity, Figure 11 shows the same data in 2d space, showing the again a general trend in increasing F1 correlating with increased instability.

4.1.3. Comparison with fully interpretable model

There are some arguments in the literature against the use of black box models, even with explainability techniques such as LIME, instead arguing that fully interpretable models are sufficient for many tasks [24]. Once the black box classifiers above were tested with LIME, an interpretable decision tree was trained for comparison. This model was able to achieve an F1 score of 0.96, lower than the black-box models which can achieve 0.98+, however it does have the advantage of being fully interpretable.

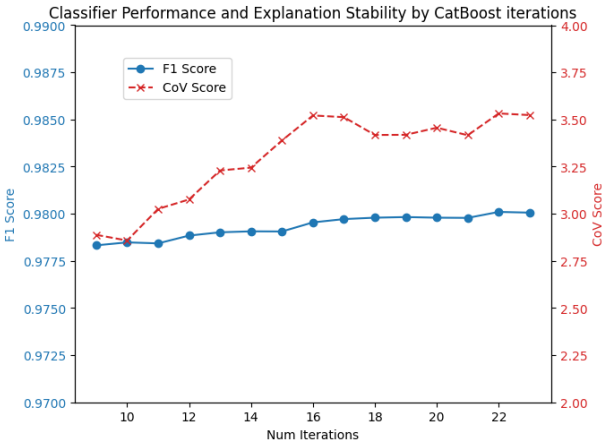

Fig. 7. F1 Score and LIME explanation stability against CatBoost num of iterations (learning rate=0.8)

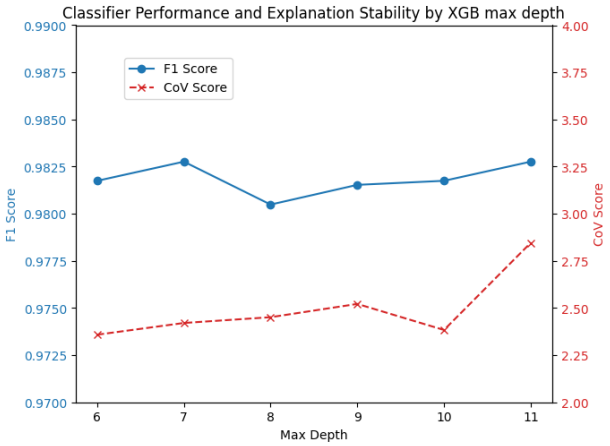

Fig. 8. F1 Score and LIME explanation stability against XGBoost max depth (eval metric = multiclass log loss, learning rate = 0.1, min child rate = 0.1)

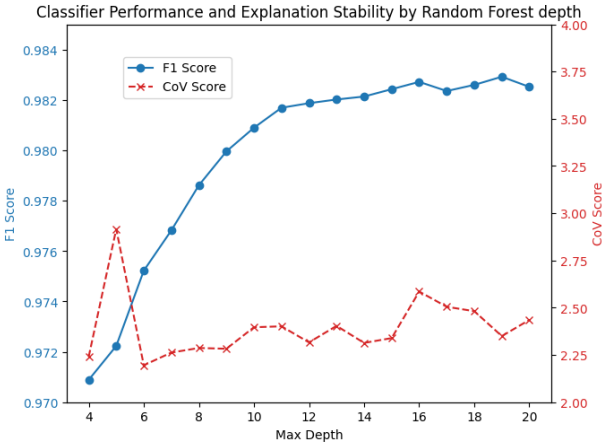

Fig. 9. F1 Score and LIME explanation stability against Random Forest max depth (min samples split = 5, min samples leaf = 1)

4.1.4. Pareto plot

The full results from each experiment can now be plotted against each other to visualise the optimisation space. Figure 12 shows the full range of solutions, Figure 13 shows the same data with the outlying poorly performing solutions removed, and the Pareto optimal solutions highlighted. The performance of the fully interpretable decision tree is shown as the line on Figure 12, to provide context for the black box models performance and show which black box models are out-performed by the interpretable model. For this dataset and experiments, the Pareto solutions are all from the random forest set of models. As is often found in multi-objective optimisation, the choice of the exact model settings will depend on the overall task and requirements of the user, therefore a final decision has not been made here. It may also be the case that where exact interpretability is very important, the user may choose to use the interpretable model with the lower F1 score. This experiment has shown however, that is possible to improve explanation stability through model parameters, and that it is worthwhile for explainable machine learning practitioners to consider this during the model selection and training stage.

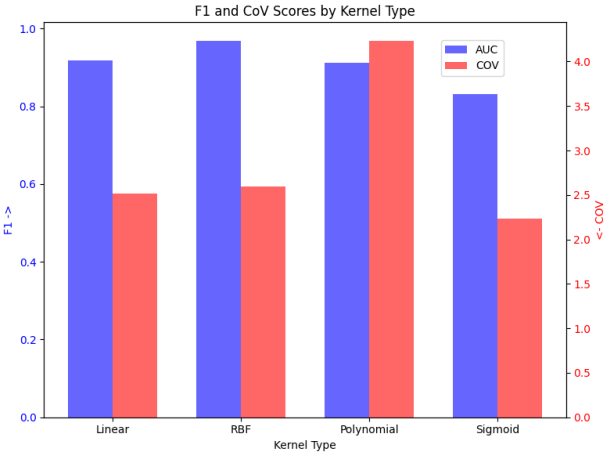

Fig. 10. Classifier performance and LIME stability for different Support Vector Classifier Kernels

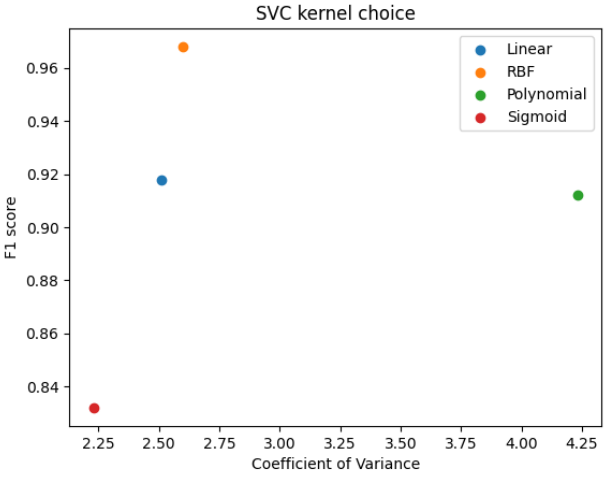

Fig. 11. Scatter plot of SVC values

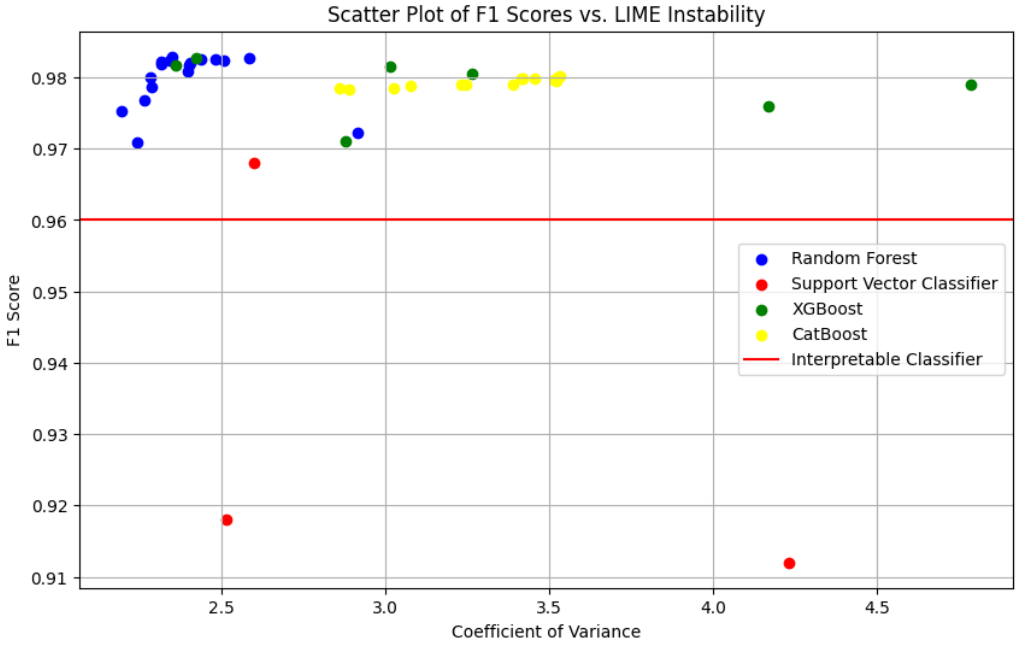

Fig. 12. Optimisation space

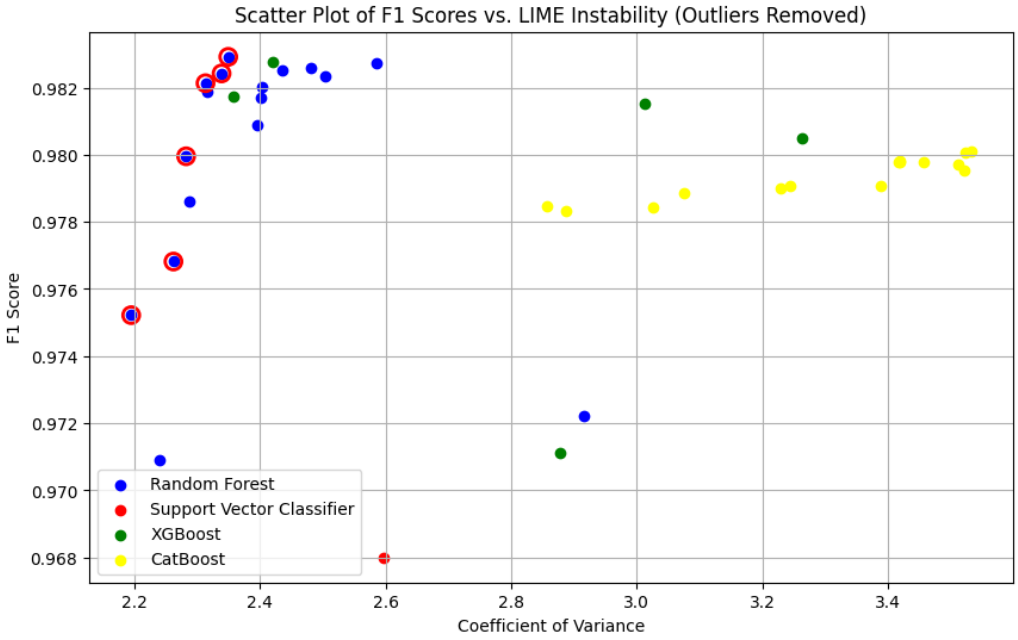

Fig. 13. Optimisation space zoomed and Pareto front highlighted

4.2. Optimisation on imbalanced dataset (credit fraud)

Imbalanced datasets are a common and difficult problem within machine learning. For this section, the highly imbalanced credit card fraud dataset [17] will be used, with only 0.017% of the data points belonging to the positive (fraudulent transaction) class. This section will investigate techniques to increase LIME stability whilst maintaining classification performance on this highly imbalanced set.

4.2.1. Baseline best classifiers

The first step was establishing the best trained models for the dataset as a baseline for experiments. Three models were used: a neural network, random forest and XGBoost.

The neural network was created using the TensorFlow library. The model uses sequential layers, with an input layer of feature size, then 16 fully connected ReLu activiation layers, a 0.5 dropout layer, then a sigmoid output layer. Early stopping was used to prevent overfitting. Optimisation used binary cross entropy and the Adam optimiser.

Figure 14 shows the confusion matrix for this models predictions. It is important to take into account the context of the dataset and the imbalanced loss requirements. A false positive (legitimate transaction blocked), may be considered less harmful than a false negative (fraudulent transaction allowed).

Figure 15 shows the confusion matrix for the random forest model. The models have similar recall for the positive class, however the random forest has much fewer false positives. XGBoost shows very similar performance in Figure 16.

Model	Accuracy	Precision	Recall	F1
Neural Network	0.9994	0.8649	0.7356	0.795
Random Forest	0.9995	0.9436	0.7701	0.8481
Extreme Gradient Boost	0.9995	0.9166	0.758	0.8301

TABLE II. Model evaluation metrics

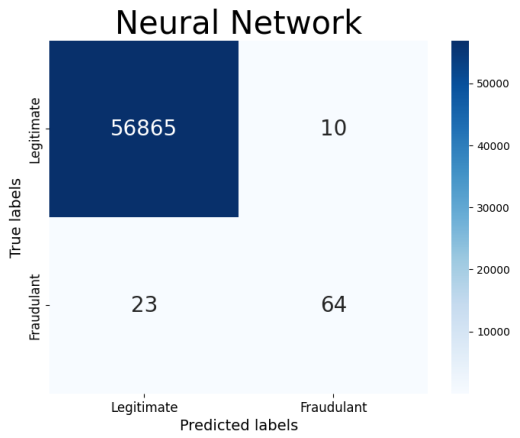

Fig. 14. Neural network confusion matrix

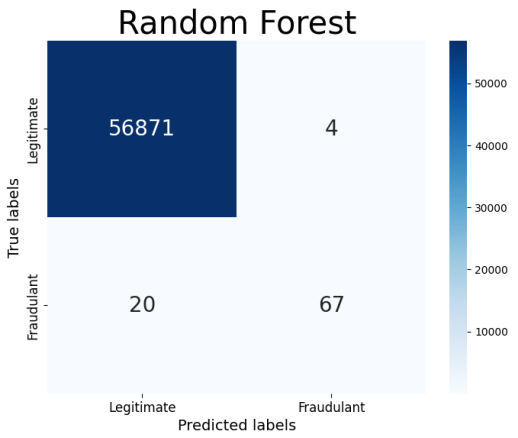

Fig. 15. Random forest confusion matrix

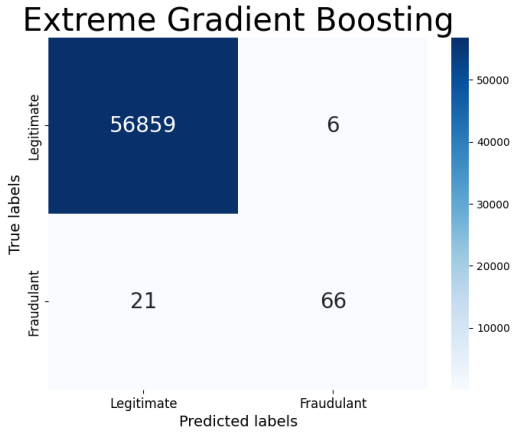

Fig. 16. Extreme Gradient Boost confusion matrix

4.2.2. Explanation stability of positive and negative class samples

The average LIME explanation stability was calculated using 30 randomly selected predictions each from class 0 and class 1 using the baseline trained classifiers, to determine whether dataset imbalance impacted the explanation stability for each class. Figure 17 shows interesting results, explanations generated on the heavily under-represented positive class are more stable than the negative class.

It is unclear why the difference exists, however there are a few possible reasons. Models trained on imbalanced data, with the appropriate loss functions to penalise misclassification of the minority class pay ”extra attention” to samples in the minority class. This may mean that the prediction space around these samples is more defined, leading to more stable predictions. The minority class samples may exhibit distinctive feature values that gives LIME more confidence in their impact, leading to higher stability. These reasons are supported by Zhang et al’s investigation into the causes of LIME instability, which found differences in stability between different data points due to the local decision boundaries and model behaviour around these points [7].

There is still plenty of potential for the stability of both classes explanations to be optimised. It is also important again to take into account the context of the dataset. It is a fair assumption that a security analyst would be more interested in explanations of positive class samples, to understand the nature of fraudulent transactions, therefore an increase in stability for the positive class is likely to yield more actionable information than an increase for the negative

class. Due to the difference in stability between the classes, experiments on this dataset will show results for the stability of each class separately.

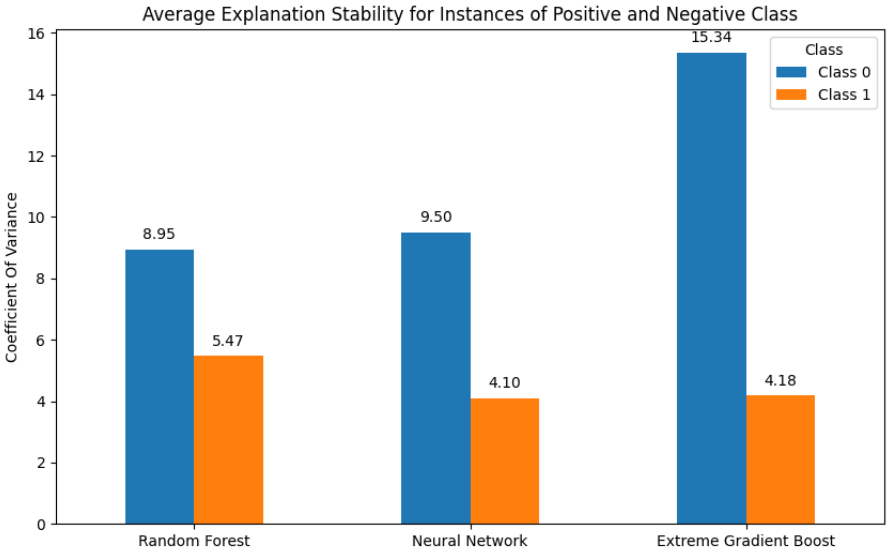

Fig. 17. Average explanation instability of positive and negative samples

4.2.3. Synthetic oversampling

A popular technique for tackling imbalanced datasets is SMOTE [25]. SMOTE generates synthetic samples from the minority class in an attempt to rebalance the dataset. SMOTE was used on the credit card dataset to generate enough samples from the minority fraudulent class to give a 50/50 class split. The models were retrained on the balanced training set, and evaluated on the original, imbalanced test set.

Figure 18 shows the LIME stability for each model and sample class when SMOTE is applied to the dataset before training. It shows encouraging results, with a decrease in instability for all models and classes, excluding Extreme Gradient Boost which saw a slight increase in instability for class 1.

Figure 19, shows the change in F1 scores for each model due to SMOTE being applied. Each model experienced a decrease to varying degrees, with the neural network suffering the most. This was due to a large increase in false positives, seen in Figure 20, compared to the baseline confusion matrix in Figure 14.

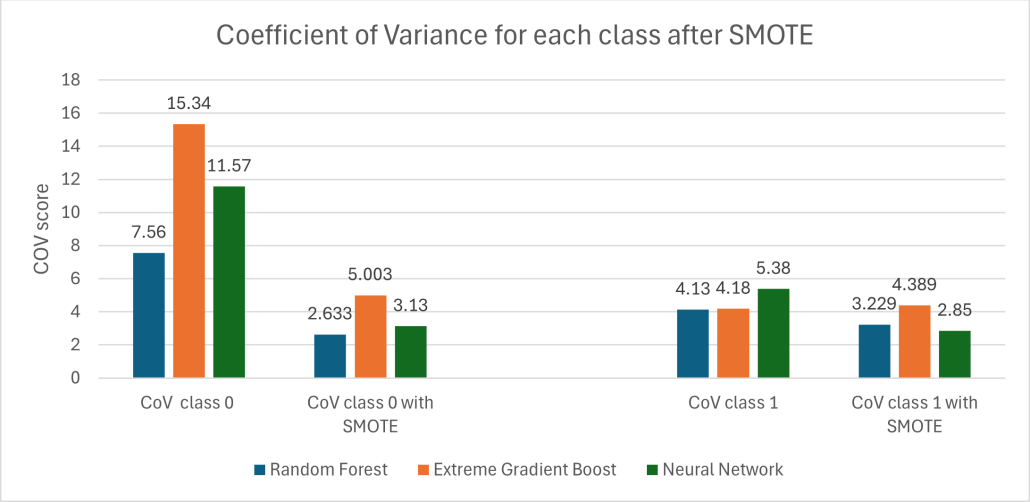

Fig. 18. Coefficient of Variance scores for LIME explanations against model/rebalancing pairs

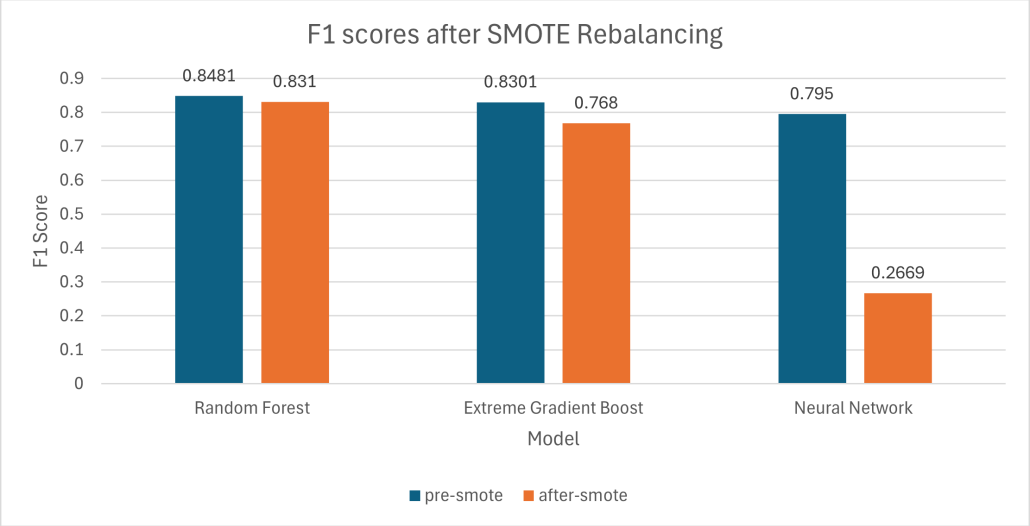

Fig. 19. F1 scores for models against model/rebalancing pairs

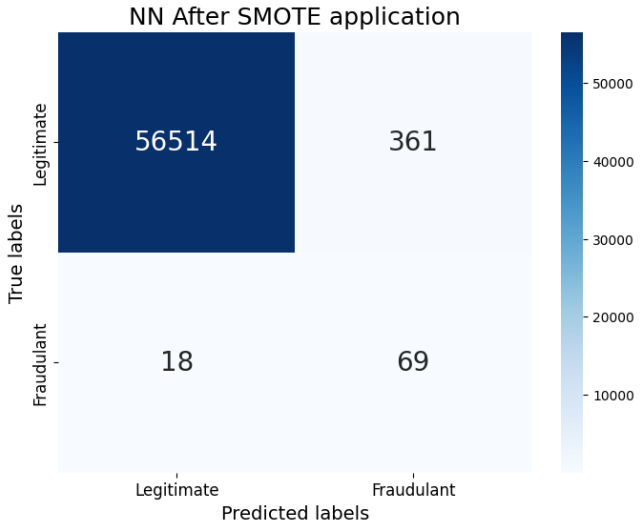

Fig. 20. Confusion Matrix for Neural Network after SMOTE application

4.2.4. Dimensionality reduction

The credit fraud dataset has a reasonably high dimensionality, with 29 features available for use in classification. This experiment investigated impact of dimensionality reduction and feature selection on classifier performance and LIME stability. Because LIME generates random samples across all dimensions, it was hoped that reducing the number of dimensions would reduce the instability when fitting LIME. To select the subset of features to use, Sequential Forward Feature Selection was used [16]. Figures 21, 22 and 23 show the impact of feature reduction. Feature reduction did not have an impact on F1 score or LIME instability, this is possibly due to the LIME algorithm performing feature selection itself through L1 regularization, hence no change in instability.

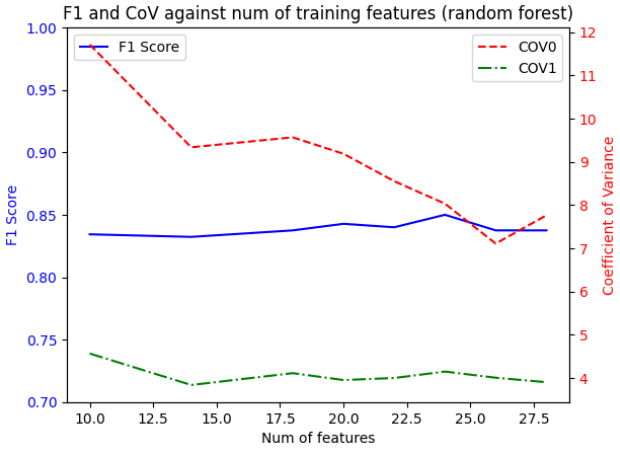

Fig. 21. Random forest feature reduction

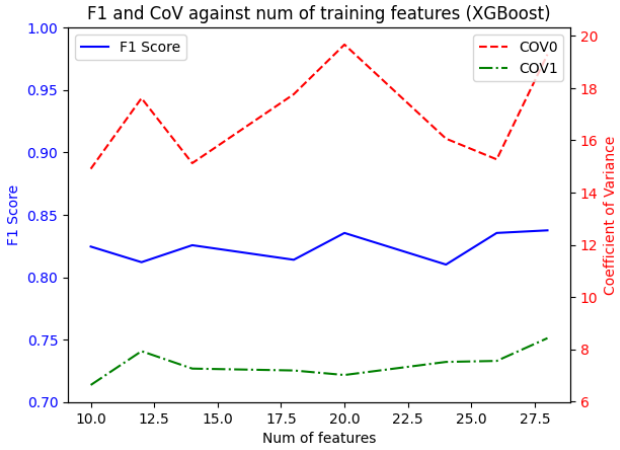

Fig. 22. XGBoost feature reduction

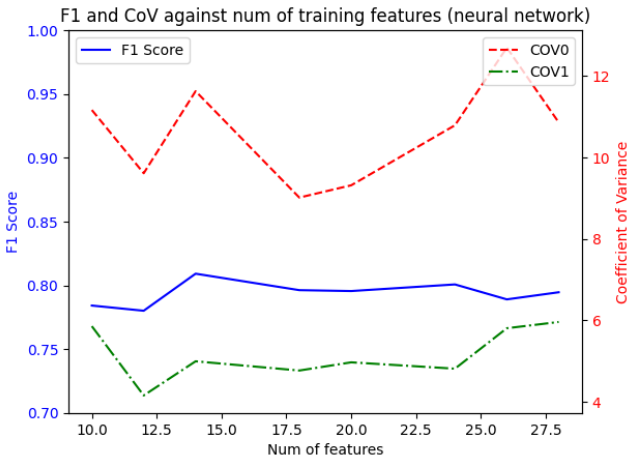

Fig. 23. Neural network feature reduction

5. Evaluation of results

These experiments have tested the relationship between black box model parameters and data preprocessing, and their impact on classifier performance and LIME explanation stability. Experiments on the DNS dataset showed a link between ensemble model depth and instability: as the model became deeper and more complex the classification score improved, however the stability of the LIME explanations were decreased. This allowed for a multi-objective optimisation space to find optimal solutions. The space showed how different models can produce very different stability results, and showed the success of the random

forest model in both classification performance and LIME stability. Within the random forest results, a set of pareto optimal solutions can be found.

Experiments on the credit fraud dataset showed promising results, with the use of synthetic rebalancing increasing LIME stability for both the majority and minority class. SMOTE rebalancing [25] has long been shown to increase classification performance on imbalanced datasets [26], this project has demonstrated that it also enhances LIME stability.

It is important to note that the specific successes in this project will not necessarily track across all datasets and situations. As with training models purely for accuracy, model choice, settings and techniques will depend heavily on the dataset used. This project however shows the potential for optimisation in this manner, and that it may be worthwhile for practitioners dealing with explainable machine learning tasks to consider their model choice and parameters to gain.

6. Critical assessment and future work

This project has investigated the potential for LIME explanation stability increase through changes in the underlying model design. Stability is vitally important for trust in the interpretability framework. An explanation that acts unpredictably over multiple runs of the same data points suggests an underlying unreliability, and doubt of the correctness of the explanations. This project has been successful in improving stability for both the DNS and credit fraud dataset, with only a limited reduction in classifier accuracy.

Stability, while vital, is not the only measure of the quality of an explanation. The accuracy and fidelity of the explanation with relation to the models actual working is very important. This measure has not been evaluated during this project, only the working assumption that a stable explanation is more likely to be correct. This is a limitation of the current impact of this project but provides an avenue for future work. With more stable explanations, it will be easier for further metrics of explanation quality to be used. Amparore et al [27] propose a set of further metrics, including local fidelity of the explanation model (how well the explanation model approximates the black box model around the chosen sample x) and conciseness of the explanation. Once stability has been achieved, these would be useful metrics to further evaluate the quality of LIME explanations.

This project has focused on tabular data, however LIME also has implementations for text and image data. These are useful future directions for research, but would require new stability metrics to be developed due to the different format that LIME returns explanations for these datatypes.

This project's scope has been solely on the LIME explanation method, however there exist many other methods such as SHAP [5] that may provide better insights, or methods that enhance LIMEs usefulness when used in conjunction. Sovrano and Vitali [28] discuss the theory of explanation in the context of explainable AI, and propose a metric to measure the usefulness of explanations based on the fundamental questions it can answer. These metrics were considered to be outside of the scope of this project due to their reliance on human surveying to understand the information gain.

7. Conclusion

Since its inception, the focus of machine learning research has been driven towards increasing the accuracy and predictive power of models, with models growing more complex and less interpretable to the person that created it. As regulation catches up and new laws are written restricting AI and ML development, it is almost certain that attention will turn to enhancing interpretability and increasing trust in black box models. One approach may be

the further development and use of LIME, by providing local explanations for individual data points showing the main features used and their contribution to a prediction or decision. This project has found that stability, and therefore the potential trust of LIME explanations can be increased through modification of the underlying classifier, with minimal loss of predictive power.

By defining a new stability metric as part of a multi-objective optimisation, optimal hyperparameters for machine learning models such as the random forest can be found. When using ensemble models, as the depth of the model increases the model accuracy increases to its peak, but the instability of the LIME explanations also increases. Mapping this problem into a multi-objective optimisation space allows for optimal tradeoff solutions to be found.

Imbalanced datasets present a number of challenges for machine learning, this project found that imbalanced data caused a disproportionate rise in LIME instability for samples in the majority class compared to the minority class. Instability for both classes was found to be reduced using SMOTE rebalancing, with only a small drop in classification performance for some models.

This project has shown that focus on the underlying model, alongside focus on the LIME algorithm is likely beneficial for improving trust in LIME predictions, and trust in machine learning as a whole. Future work includes applying these principles to a wider range of datasets, models and evaluating the quality of LIME explanations using further metrics.

---

[1] T. Telford, “Apple card algorithm sparks gender bias allegations against goldman sachs,” *Washington Post*, vol. 11, 2019, [Accessed 23/04/24]. [Online]. Available: https://www.washingtonpost.com/business/2019/11/11/apple-card-algorithm-sparks-gender-bias-allegations-against-goldman-sachs/

[2] “Regulation of the European Parliament and of the Council on laying down harmonised rules on Artificial Intelligence (Artificial Intelligence Act) and amending certain Union Legislative Acts,” Official Journal of the European Union, European Parliament and Council of the European Union, 2024.

[3] House of Lords, “Artificial intelligence (regulation) bill,” Nov 2023, [Accessed 18/03/24]. [Online]. Available: https://bills.parliament.uk/publications/53068/documents/4030

[4] C. Molnar, *Interpretable Machine Learning*, 2nd ed., 2022, [Accessed 12/02/24]. [Online]. Available: https://christophm.github.io/interpretable-ml-book

[5] S. M. Lundberg and S.-I. Lee, “A unified approach to interpreting model predictions,” in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, ser. NIPS’17. Red Hook, NY, USA: Curran Associates Inc., 2017, p. 4768–4777.

[6] M. T. Ribeiro, S. Singh, and C. Guestrin, “”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*, ser. KDD ’16. New York, NY, USA: Association for Computing Machinery, 2016, p. 1135–1144.

[7] Y. Zhang, K. Song, Y. Sun, S. Tan, and M. Udell, “”why should you trust my explanation?” understanding uncertainty in lime explanations,” 2019, [Accessed 04/04/23]. [Online]. Available: https://arxiv.org/abs/1904.12991

[8] Z. Zhou, G. Hooker, and F. Wang, “S-lime: Stabilized-lime for model explanation,” in *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery amp; Data Mining*, ser. KDD ’21. ACM, Aug. 2021.

[9] G. Visani, E. Bagli, and F. Chesani, “Optilime: Optimized lime explanations for diagnostic computer algorithms,” 2022, [Accessed 10/03/24]. [Online]. Available: https://arxiv.org/abs/2006.05714

[10] M. R. Zafar and N. Khan, “Deterministic local interpretable model-agnostic explanations for stable explainability,” *Machine Learning and Knowledge Extraction*, vol. 3, no. 3, pp. 525–541, 2021.

[11] G. Visani, E. Bagli, F. Chesani, A. Poluzzi, and D. Capuzzo, “Statistical stability indices for lime: Obtaining

reliable explanations for machine learning models," *Journal of the Operational Research Society*, vol. 73, no. 1, p. 91–101, Feb. 2021.

[12] C. Marques, S. Malta, and J. P. Magalhães, "Dns dataset for malicious domains detection," *Data in Brief*, vol. 38, p. 107342, 2021.

[13] M. Antonakakis, R. Perdisci, D. Dagon, W. Lee, and N. Feamster, "Building a dynamic reputation system for dns," *Proceedings of the 19th USENIX Security Symposium*, pp. 273–290, 09 2010.

[14] Y. Zhauniarovich, I. Khalil, T. Yu, and M. Dacier, "A survey on malicious domains detection through dns data analysis," *ACM Computing Surveys*, vol. 51, no. 4, p. 1–36, Jul. 2018.

[15] N. Aslam, I. U. Khan, S. Mirza, A. AlOwayed, F. M. Anis, R. M. Aljuaid, and R. Baageel, "Interpretable machine learning models for malicious domains detection using explainable artificial intelligence (xai)," *Sustainability*, vol. 14, no. 12, 2022.

[16] P. Pudil, J. Novovičová, and J. Kittler, "Floating search methods in feature selection," *Pattern Recognition Letters*, vol. 15, no. 11, pp. 1119–1125, 1994.

[17] Machine Learning Group - Université Libre de Bruxelles, "Credit card fraud detection," Kaggle dataset, 2013, [Accessed 01/04/24]. [Online]. Available: https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud

[18] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, Oct 2001.

[19] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '16. ACM, Aug. 2016.

[20] L. Prokhorenkova, G. Gusev, A. Vorobev, A. V. Dorogush, and A. Gulin, "CatBoost: unbiased boosting with categorical features," 2019, [Accessed 02/04/24]. [Online]. Available: https://arxiv.org/abs/1706.09516

[21] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A training algorithm for optimal margin classifiers," in *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, ser. COLT '92. New York, NY, USA: Association for Computing Machinery, 1992, p. 144–152.

[22] T. Chen, C. Guestrin *et al.*, "Xgboost: Scalable and flexible gradient boosting," Github Repository, 2019, [Accessed: 2023-04-30]. [Online]. Available: https://github.com/dmlc/xgboost

[23] "Catboost: Gradient boosting on decision trees library by yandex," Github Repository, 2017, [Accessed: 2023-04-30]. [Online]. Available: https://github.com/catboost/catboost

[24] C. Rudin, "Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead," *Nature Machine Intelligence*, vol. 1, pp. 206–215, 2019.

[25] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: Synthetic minority over-sampling technique," *Journal of artificial intelligence research*, vol. 16, pp. 321–357, 2002.

[26] A. Fernández, S. García, N. V. Chawla, and F. Herrera, "Smote for learning from imbalanced data: Progress and challenges," *Journal of Artificial Intelligence Research*, vol. 61, pp. 863–905, 2018.

[27] E. Amparore, A. Perotti, and P. Bajardi, "To trust or not to trust an explanation: using leaf to evaluate local linear xai methods," *PeerJ Computer Science*, vol. 7, p. e479, Apr. 2021.

[28] F. Sovrano and F. Vitali, "An objective metric for explainable ai: How and why to estimate the degree of explainability," *Knowledge-Based Systems*, vol. 278, p. 110866, 2023.