Machine Learning Data Suitability and Performance Testing Using Fault Injection Testing Framework

Manal Rahal[1][0009−0006−7733−8298], Bestoun S. Ahmed[1,2][0000−0001−9051−7609], and Jörgen Samuelsson[3][0000−0003−1819−1709]

[1] Dept of Mathematics and Computer Science, Karlstad University
manal.rahal@kau.se
[2] Department of Computer Science, Faculty of Electrical Engineering, Czech Technical University in Prague, Prague, 16627, Czech Republic
bestoun@kau.se
[3] Dept of Engineering and Chemical Sciences, Karlstad University
jorgen.samuelsson@kau.se

Abstract. Creating resilient machine learning (ML) systems has become necessary to ensure production-ready ML systems that acquire user confidence seamlessly. The quality of the input data and the model highly influence the successful end-to-end testing in data-sensitive systems. However, the testing approaches of input data are not as systematic and are few compared to model testing. To address this gap, this paper presents the Fault Injection for Undesirable Learning in input Data (FIUL-Data) testing framework that tests the resilience of ML models to multiple intentionally-triggered data faults. Data mutators explore vulnerabilities of ML systems against the effects of different fault injections. The proposed framework is designed based on three main ideas: The mutators are not random; one data mutator is applied at an instance of time, and the selected ML models are optimized beforehand. This paper evaluates the FIUL-Data framework using data from analytical chemistry, comprising retention time measurements of anti-sense oligonucleotide. Empirical evaluation is carried out in a two-step process in which the responses of selected ML models to data mutation are analyzed individually and then compared with each other. The results show that the FIUL-Data framework allows the evaluation of the resilience of ML models. In most experiments cases, ML models show higher resilience at larger training datasets, where gradient boost performed better than support vector regression in smaller training sets. Overall, the mean squared error metric is useful in evaluating the resilience of models due to its higher sensitivity to data mutation.

Keywords: Mutation testing · Data mutation · Fault injection · Machine Learning Testing · Responsible AI · Chromatography data

1   Introduction

The world is experiencing rapid evolution in using artificial intelligence and machine learning (ML) in almost every domain. This trend has raised questions about the resilience of ML systems to data faults, most importantly in safety-critical applications such as autonomous driving, cyber security, and healthcare [30]. In light of these concerns and the serious consequences of failure in ML systems, testing methods have gained much attention in the research community and industry [18]. Furthermore, the resilience of ML systems has become an important requirement to gain users' trust [9]. Given its unpredictable behavior, testing ML systems is more complex than classical software since inspecting the ML algorithm alone is not sufficient [2, 20]. The behavior of the ML system does not depend solely on the algorithm but also on the training and testing data, the choice of hyperparameters, and the optimizer. All these factors impact the performance of the system [23]. The influence of data on the performance of the ML system is not negotiable. Therefore, performing tests to evaluate the suitability of the data is equally critical to achieving a production-ready ML system.

In the literature, various systematic testing methods are effective in testing the ML model, such as mutation testing (MT) and black-box testing tools [19]. However, not as many systematic methods were investigated to evaluate the training data as stated by Narayanan *et al.* [19]. Having an important impact on the performance of the ML, major faults in the input data could lead to incorrect outcomes by the system. In some cases, not faults, but drifts in the real-time data lead to undesirable outcomes. Fault injection is one of these few available methods that aim to intentionally inject faults into the data, as described by [23] as data sensitive faults, in an attempt to change the behavior of the system [6]. The more an ML system is resilient to data-sensitive faults, the better a system can learn from incomplete data and unexpected observations. Therefore, models that have high resilience generally generalize better to far-from-perfect real-world data.

ML systems based on supervised learning algorithms often assume the input data is static. However, this assumption does not necessarily hold when the system is deployed in the real world [11]. Therefore, unexpected events are likely to occur and can cause risks to the performance of the ML system. But a resilient ML system, as described in [28], has the capacity to absorb data fluctuations without performance degradation. According to [28], a resilient system can monitor, learn, anticipate, and respond to adversity. This means that the system should be able to maintain good performance when the input data are disrupted, to a certain extent. The sources of data disruption are many and could be classified as natural, system-related, or human errors, and can also include external factors [28]. The diversity in the types of faults raises questions such as are some ML models more resilient to data faults than others and which faults have the biggest influence?

Like any ML system, multiple fault sources can influence the data collected from a chromatography system. Consistent records of data could be attributed to

the performance of the chromatographic instruments, experimental conditions, and other external factors. Such variations and errors are common in any real-world application and lead to degradation in the performance of the ML system. Therefore, there is a need for ML models that can predict under complicated uncertainty, yet perform efficiently when used in decision-making [16]. To address this gap, we present the Fault Injection for Undesirable Learning in input Data (FIUL-Data) testing framework to evaluate the resilience of an ML system to common faults. The design intends to introduce likely-occurring faults through artificial mutators before applying the ML model. The proposed framework is generalizable, explainable, feasible on a scale, and applicable in multiple domains. The main ideas behind designing the FIUL-data framework are (1) the mutators are not random; they are formulated based on previous knowledge about the data, (2) it is a single fault application; so that at any instance of time only one fault is applied, and (3) the ML models used in the evaluation phase are selected based on their suitability to the dataset. FIUL-Data empirically validates the resilience of ML systems by introducing data faults to the ML input data in scenarios that might otherwise be rarely considered. FIUL-Data framework is evaluated on a case study from the analytical chemistry domain. The data set includes antisense oligonucleotide sequences (ASOs) and their experimentally observed retention time (t_R). These text-coded sequences are transformed into numeric features before applying any ML system [31]. The data set is collected through a sequence of chromatography experiments from the Chemistry Department of Karlstad University. The evaluation of FIUL-Data consists of two steps; first, each ML model is evaluated individually. In the second step, the models are compared with each other to have a comparative view of the vulnerabilities of each model against the different fault injections. We are mainly considering supervised ML models in our case study. However, the proposed framework could apply to other types of ML systems.

In this paper, we propose and evaluate the FIUL-Data framework usable in multiple domains against likely occurring data faults. As a result, this paper makes the following contributions:

- Propose FIUL-Data as a data-mutation-based framework integrated with ML to evaluate the resilience of ML systems to data faults.
- Design and implement two data-level mutators applied to introduce likely-occurring faults to ML input data.
- The proposed FIUL-Data framework is evaluated on a dataset from the analytical chemistry field to demonstrate the usefulness of the framework in a real-world application.
- Propose and perform a multi-metric evaluation of the FIUL-Data framework to enable quantitative evaluation of the metrics and generation of insights.

The remainder of this paper is structured as follows: Section 2 summarizes the relevant background concepts and lays out the necessary terminologies to understand the paper. Section 3 details the research questions and the experimental setup to apply and evaluate the proposed framework. The evaluation

results from the use case and the answers to the research questions are presented in Section 4. Finally, Section 5 concludes the paper with a summary of the findings.

2   Background

This section provides a selected overview of relevant MT definitions and applications in the literature.

2.1   ML in Chromatography Applications

Chromatography is an important separation method that is used in all chemistry fields [5]. Chromatography is considered powerful in separating mixtures of compounds even with similar physical properties due to the large number of partitioning steps involved [5]. The output of the separation is gaussian-shaped peaks for each eluted compound in the mixture. ML is commonly used in chromatographic separation applications to predict experiments before they are conducted in the laboratory. This is possible by predicting important parameters, such as t_R, which is the time a compound spends in the system from injection to elution. One of the most important goals in chromatographic separation is to achieve a sufficiently high resolution between the eluted peaks within a reasonable experimental time and resources [13]. Therefore, optimizing the experiment conditions to achieve this goal requires much effort. However, the optimization task can be complicated and time-consuming, given the large space of experimental variables and the possibilities of interaction among them [29]. The accurate performance of ML predictive models allows analytical chemists to reduce the costly and time-consuming experiments needed to achieve optimal separation conditions. In such cases, peak resolutions or t_R could be predicted as the output. Once the space of chemical conditions is controlled, more efficient separation experiments can be conducted in the laboratory. In the literature, various ML models have been tested in this context, including artificial neural networks (ANN) and traditional models such as in [4, 12, 25], which have shown promising results.

Another important application of ML in chromatography is to use it to analyze the chromatographic data that result from the experiments. Regardless of the type of chromatography system, data analysis is time-consuming and often requires manual human intervention [24]. Therefore, researchers actively investigate the potential of ML to reduce human-dependent steps in the analysis stage using mainly ANN, as in [3,22,24,27]. For example, [24] tested an approach based on convolutional neural networks to automatically evaluate the modeled elution profiles of the gas-chromatographic data. In [3], a multilayer neural network (NN) was applied to predict the retention behavior of amino acids in reversed-phase liquid chromatography. The authors concluded that NNs are powerful in modeling the influence of various gradient elution modes. ML offers great potential to advance the separation tasks toward more efficient handling of chromatographic data from collection to analysis. In this paper, we provide another example of

the usefulness of ML with a focus on building more resilient separation pipelines in chromatography.

2.2 Common Faults in Chromatography Data

The reliability of the data generated by chromatography experiments has been investigated greatly in the literature. Although modern instruments used in laboratories have user-friendly interfaces, some types of variation and errors are inevitable [15]. Multiple approaches have been proposed to classify and quantify the errors in an attempt to understand their influence on the certainty of the measurements. In general, variations can be seen as person or system caused. Guiochon *et al.* [1] discuss the different sources of errors and their influence on chromatographic measurements. Kuselman *et al.* [14] identify nine human-related errors while performing experiments. The most common sources of common systematic errors in chromatography were also investigated in [10]. The causes of systematic errors ranged from errors in sample handling to wrong evaluation and interpretation of results. In the case of ASO chromatography experiments, errors are reflected in the output data in the form of skewness in the detected peaks, low signal-to-noise ratio, and high variation of the outcome across replicates of the sample. The inevitability and recurrence of errors in chromatography, demand handling before data are used for analysis or ML purposes [8]. MT methods integrated with ML are one of the approaches used to simulate likely-to-happen errors and study their influence on the performance of the models.

2.3 Data Mutation

To ensure the systems work as intended in real but uncertain conditions, we must understand and consider the faults in our input data [20]. There is no doubt that different types of faults would have a different influence on the performance of the ML system. Many approaches deal with the different faults, such as changing the input data, changing the model, or building a resilient system to faults [20]. One of the approaches to understand the influence of certain errors is the application of designed artificial mutations to the input data.

Originally, MT is a popular software engineering (SE) technique widely used in academia and industry [21]. The concept of MT is based on using artificial faults for system testing purposes. In other words, MT for evaluation is used to measure the effectiveness of a system in finding faults [32]. Traditionally, MT application involved inserting individual faults into the software, but later approaches tested higher-order mutations in which multiple faults are injected at once, which was shown to be expensive [7]. Although MT is a proven technique in SE, it recently started gaining attention in the field of ML, specifically in the subfield of deep learning (DL) [26]. However, MT for ML systems is still considered to be in the early stages [17]. The interest in adopting established methods such as MT is triggered by the willingness to improve the trustworthiness of DL

systems [26]. It should be noted that by concept, MT has been most commonly used to test models, but rarely at the data level, which is our paper's focus.

Motivated by the success of MT in classical software systems and recently in DL models, the FIUL-Data framework applies mutation at the data level. In the literature, the tools used to evaluate the input data for ML are limited [19]. To address this gap, we propose the FIUL-Data framework to assess the responsiveness of multiple ML models to input data mutations. The framework is applied to ASO chromatographic data where the designed data mutations represent common faults in the field.

3   Methodology

In this section, we describe the systematic approach followed to study the resilience of ML models to data faults, including the specific research questions that guided the experiments. The dataset used to evaluate the FIUL-Data framework and the methods used for the evaluation are also described in detail.

3.1   Research Questions

This paper aims to answer the following research questions (RQs):

- **RQ1: How does the reduction in training data influence the resilience of the ML model?**
- **RQ2: How does the selection of the size of a certain class of data influence the resilience of the ML model?**
- **RQ3: How to evaluate the resilience of different ML models in response to different data mutators?**

3.2   Use Case Dataset

The data used in our use case are obtained by means of a chromatography experiment aimed at separating impurities from an ASO compound. The experiments were carried out under two different chemical conditions, therefore resulting in two different datasets G1 and G3. During the separation process, an aqueous-organic mixture is continuously pumped into the chromatography column, where the amount of organic solvent in the mixture increases over time (gradient time); this is called the gradient. The first dataset (G1) is collected from experiments in which the gradient equals 11 minutes. The second dataset (G3) is collected at a gradient of 44 minutes. The change in gradient results in a new t_R for each unique compound entering the chromatography system. At higher gradients, ASO compounds are retained longer in the system. Therefore, data collected at higher gradients are considered more sensitive to slight changes in experimental conditions.

In chemistry, the ASO compound is represented in a combination of four different nucleotide bases, adenine (A), thymine (T), cytosine (C), and guanine

(G), forming a sequence. In this case study, non-phosphorothioated known as native and phosphorothioated ASO sequences are collected in a dataset with their respective t_R as the target variable. The t_R is always recorded as the separated compounds individually exit the system.

After pre-processing the datasets and removing the incomplete records, the clean G1 dataset included 876 data points and G3 dataset had 870 data points. Both datasets have compounds that do not include sulfur (non- phosphorothioated), partially phosphorothioated compounds, and others fully phosphorothioated. Both datasets have more than 79% of the compounds partially or fully phosphorothioated. As part of the data preparation methods before applying ML, the nucleotide sequences are encoded into numeric values, referred to as features, such as the frequency of each nucleotide and di-nucleotide (ordered and unordered) in a sequence, the total length, and the number of sulfur atoms present. Figure 1 shows the frequency range for the encoded features in the G3 training dataset. The same encoding system is applied to the testing data. As seen in Figure 1, the ASO sequence can reach 20 nucleotides long, whereas, the number of sulfur atoms in the sequences varies between 0 and 19 atoms. The occurrence of A, C, T, and G nucleotide bases is relatively similar, while, in di-nucleotide occurrence, TT and CC are the most frequent.
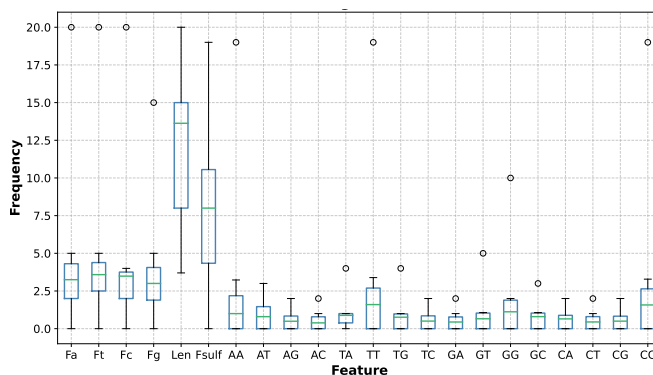

Fig. 1: Frequency of features in G3 training data. The features representing unordered di-nucleotides were removed for visualization purposes.

3.3 The FIUL-Data Framework

The FIUL-Data framework is built on four key phases, as illustrated in Figure 2. In the first phase, the data mutators are designed and coded. The operation of the data mutators is application-specific and relies on the common faults encountered during implementation. The execution of data mutators comes next, where pre-trained ML models and the programmed data mutators are imported and performed on clean data. Once the mutated data are ready, the ML cycle

starts, including typical training and evaluation of the models based on consciously selected metrics. The results are integrated into useful visualizations in the last phase, and insights are concluded. In our case study, Python language is selected to implement the data mutators' functions and run the experiments.
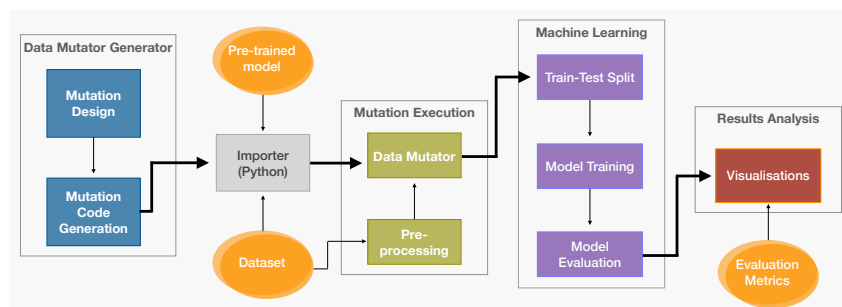

Fig. 2: Conceptual illustration of the FIUL-Data framework

The application of FIUL-Data framework consists of two general-purpose data mutators, the reduce_data_mutator and the select_data_mutator. The operation of both data mutators is described in detail in the following section. We note that the same data mutators are expected to behave differently in different applications. In this application, 10% of the data is repeatedly removed in every iteration until only 10% of the original data is left. The selected step-size induces small changes in the data that allows to record results and visualize behavioral trends from 10 data mutation iterations. In other words, the 10% iterative variation ensures that we have a sufficient number of instances from small yet sensitive changes that allows to monitor the behavior of the models. We note that the records in the dataset are randomly shuffled before applying the FIUL-Data framework. This is an important step, as compounds and impurities could share similar characteristics in the case of chromatography data. Data shuffling ensures representative distribution of the different compounds in the partitions of the training and testing sets.

– **Reduce_data_mutator** The reduce_data_mutator reduces the training data by 10% in each iteration. In some applications, where collecting data for ML is expensive and time-consuming, it is critical to know the size of training data that is sufficient for a good-performing model. Large data are a relative term and depend on the application being studied. In the case of chromatography data, laboratory experiments are expensive, the products used to perform the experiments are costly, and the experiments run for a long time. Therefore, collecting sufficient data could significantly reduce time and cost burdens. In our experiment, the data is first split into train and test datasets, 80% and 20%, respectively. The testing data remains unchanged, while the training data is reduced iteratively by 10%. For every iteration,

the model is fitted to the training data and t_R is predicted on the unseen data. At the end of each iteration, the coefficient of determination (R^2) train, R^2 test, and the mean squared error (MSE) are recorded. The design and operation of reduce_data_mutator is illustrated in Figure 3.

– **Select_data_mutator** The select_data_mutator iteratively reduces a certain class in the data. In this case, the records of the compounds that lost one or more sulfur atoms from their sequence are removed at a rate of 10% per iteration, while the other class (native compounds) remains unchanged. However, the percentage of the target class in the training and testing data after splitting is controlled for consistency purposes. The sulfur atom(s) loss is denoted by "-P=O" at the end of an ASO sequence. The models are then fitted to the new version of the training data and evaluated on the testing data. The operation of select_data_mutator is illustrated in Figure 4. This mutator aims to reveal the influence of a certain class of data on the performance of the ML system. In this case study, the class of sequences having a -P=O suffix is subject to data mutation.
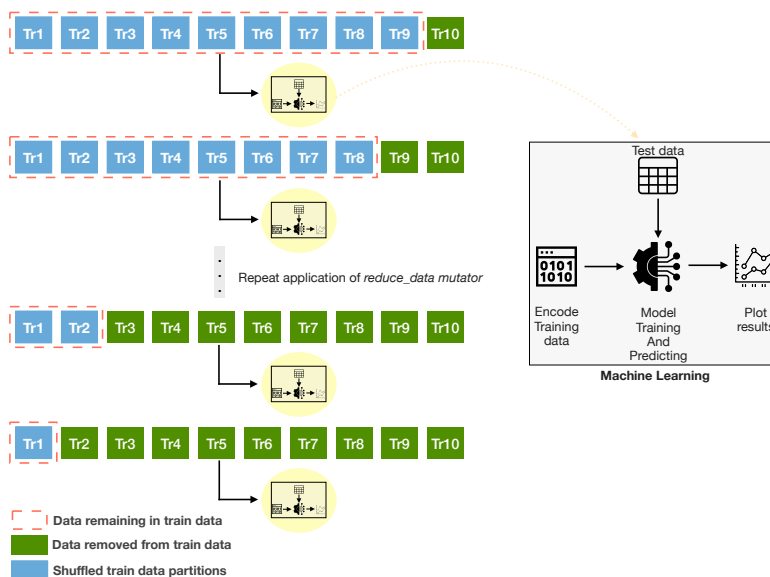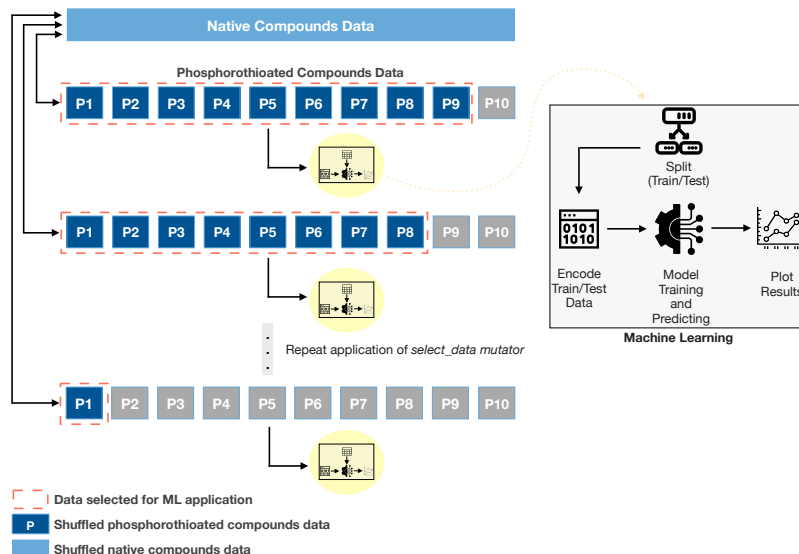

Fig. 3: The operation of reduce_data_mutator

4   Results and Analysis

After implementing the reduce_data_mutator and the select_data_mutator functions in a Python (3.9) supported framework, the G1 and G3 datasets along

Fig. 4: The operation of select_data_mutator

with the corresponding pre-trained and hypertuned ML models, were imported. Then, the FIUL-Data framework was applied to both datasets, where the performances of the Gradient boost(GB) and support vector regression (SVR) models in response to the type of fault injected were compared. The evaluation metrics used in monitoring the performance of individual models and when comparing models to each other were the MSE, the R^2 train, and the R^2 test. The MSE is chosen to observe the variation in the average squared difference between the predicted values and the observed values of t_R.

4.1   Effect on G1 Dataset

In Figure 6, the reduce_data_mutator is applied to the G1 dataset at a decreasing rate of 10% in each iteration. For each of the mutation iterations, the accuracy of the train and the test are recorded in addition to the MSE values. The performance of the SVR and GB models on G1 data yields relatively good results. The accuracy on unseen data for the GB model ranges from 0.80 to 0.82, showing relatively stable performance against reducing the size of the training data. For the SVR model, the R^2 test remained relatively stable until 60% of the training data were removed, and the performance began to degrade, reaching a minimum R^2 test of 0.74. Both models were shown to generalize reasonably well to unseen testing data when the training data is relatively large; however, GB performed better with a smaller training datasets. The same trend applies to MSE, which increased significantly for the SVR model with decreasing training data.
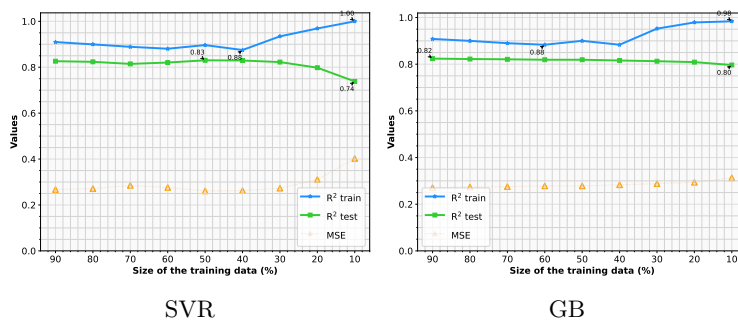
Fig. 6: The application of reduce_data_mutator in G1 dataset using SVR and GB models. Maximum and Minimum values are annotated.

The results of the application of select_data_mutator to input data are shown in Figure 8. For the GB model, we can observe that the highest R^2 test is achieved with a 30% (0.98) subset size of the -P=O class. The R^2 test ranges from 0.86 to 0.98 and the graph shows a trend of increasing R^2 test with a decrease in the subset size of the -P=O data class. The GB and SVR models performed relatively similarly, with GB showing slightly better performance across iterations. However, high fluctuations are observed in the performance of both models in the case of select_data_mutator. The same trends are observed in the MSE values, ranging from 0.017 to 0.203 in GB and 0.018 to 0.029 in SVR. MSE is a measurement of error, so the lower the MSE value, the better the performance of the model. Both models achieved the lowest performance at the sizes of the subsets 90% and 30% of the -P=O class.
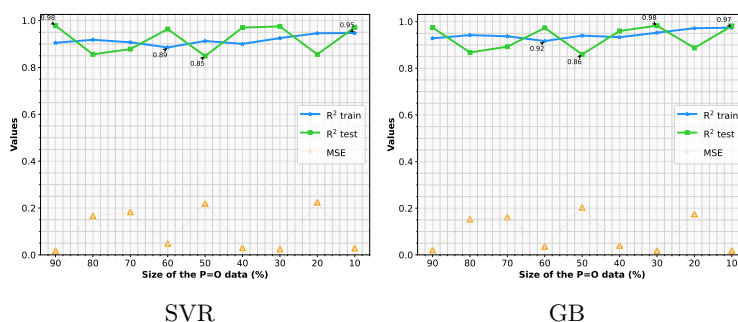

Fig. 8: The application of select_data_mutator in G1 dataset using SVR and GB models. Maximum and Minimum values are annotated.

4.2   Effect on G3 Dataset

Across both mutation applications and in almost all iterations, GB outperformed SVR as shown in the MSE trend line in Figures 10 and 12. Starting with the results of reduce_data_mutator, both GB and SVR models showed relatively high performance on unseen data, with the R^2 test reaching a maximum value of 0.92 and 0.91 respectively. The R^2 train for both models is consistently high, indicating effective learning during the training process. After 60% of the training data is reduced, both models start to show over-fitting behavior where the training performance is exceptionally high, unlike the degrading R^2 test values. The R^2 test in both models remains stable until 70% of the training data is removed, where the R^2 test begins to show a downward trend. This trend is also reflected in the behavior of MSE, where it first shows a consistent trend, then a significant increase is observed after the 7th iteration of the data mutation.
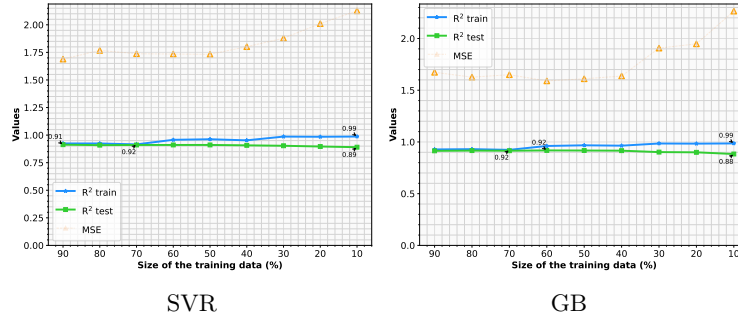

Fig. 10: The application of reduce_data_mutator in G3 dataset using SVR and GB models. Maximum and Minimum values are annotated.
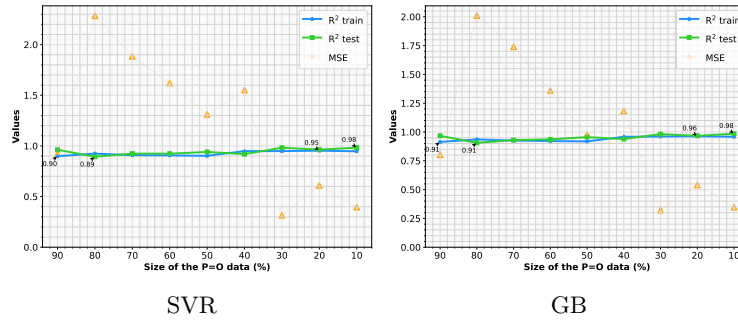

Fig. 12: The application of select_data_mutator in G3 dataset using SVR and GB models. Maximum and Minimum values are annotated.

Figure 12 illustrates the results of the application of the select_data_mutator in G3 dataset. The values of R^2 train and R^2 test in the GB and SVR models are relatively similar and show stable behavior. However, a significant decrease in MSE is observed after the 2nd iteration, reaching 1.31 and 0.9 in SVR and GB, respectively. Another steep decline is observed after the 6th iteration where the error value decreases by 79.7% and 73.7% in SVR and GB, respectively.

4.3 Effect Analysis

For a coherent comparison of the results among data mutators and models, the fluctuation trends in the R^2 test and the MSE values are visualized in Figures 14, 16. Section 4.3 is structured to answer the RQs 1-3 presented in Section 3.1

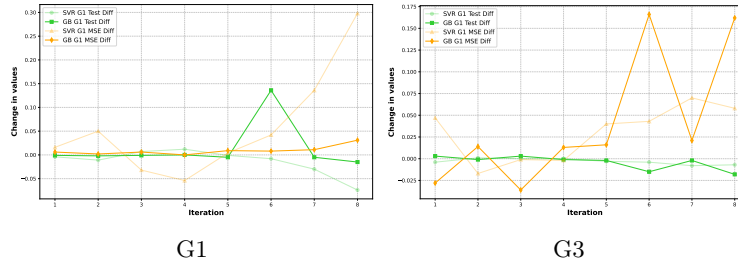G1   G3

Fig. 14: Reduce_data_mutator comparison results in G1 and G3 datasets. Iterations of data mutation is on x-axis, and relative change in values of R^2 test and MSE on y-axis.

RQ1: How does the reduction in training data influence the resilience of the ML model?

Despite some similarities in the behavior of GB and SVR, in both datasets G1 and G3, GB performed better and showed greater resilience to the reduction in the mutation of the training data. The highest R^2 test and lowest MSE were attained at the 6th iteration indicating that, for ASOs data, collecting a large training dataset does not necessarily improve the performance of the ML model. In this case, 30% of the available data were sufficient to achieve the best performance.

In the case of G3 dataset, which is a more noisy and volatile dataset, the GB model showed high resilience in the predicted R^2 test; however, after the 5th iteration, the MSE fluctuations recorded slightly higher changes. Both models performed closely until the 5th iteration, where 50% of the training data was removed. In subsequent iterations, the models showed an increase in MSE and a spiking pattern in the case of the SVR model, as shown in 14. The obtained results are expected since the ML model needs more data to be able to sustain good performance in noisy data and generalize without fitting the noise.

RQ2: How does the selection of the size of a certain class of data influence the resilience of the ML model?

In response to the select_data_mutator both models seem to have identical fluctuating behavior in the MSE values. The change in the R^2 test values is nearly negligible. As a result, in the G1 dataset, both models show similar unpredictable performance in response to the decrease in one class of the data.

Despite the fluctuating performance in response to the reduction of a specific class from the ASOs data, GB model showed slightly better resilience corresponding to the MSE pattern across iterations in G3 data. The resilience of the models during the first four iterations shows relatively stable behavior, contrary to smaller datasets.

RQ3: How to evaluate the resilience of different ML models in response to different data mutators?

The fluctuations in MSE values in response to the reduce_ and select_data_ mutators were more significant than that of the R^2 test as shown in Figure 14 and Figure 16. Therefore, MSE shows higher sensitivity to data mutation changes compared to other investigated metrics.
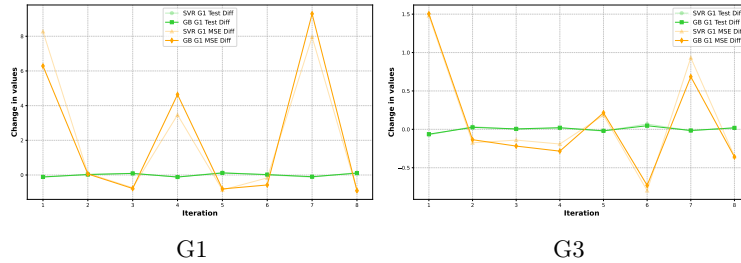

G1   G3

Fig. 16: Select_data_mutator comparison results in G1 and G3 datasets. Iterations of data mutation is on x-axis, and relative change in values of R^2 test and MSE on y-axis.

5   Conclusion

To test the resilience of ML models to multiple intentionally triggered faults, we present a Fault Injection for Undesirable Learning in the input data (FIUL-Data) testing framework. The proposed framework is evaluated on a case study of ASOs data where the performance of GB and SVR models is compared for each data mutator. In response to reduce_data_mutator, both models show relatively high resilience in larger datasets with GB outperforming SVR. We observe that 30% of the available data were sufficient to achieve the best performance. Regarding the second type of mutation, the models show greater resilience in

the G3 dataset with a decreasing trend of MSE compared to G1, which had unpredictable performance. This shows that when the size of the -P=O data is smaller, an ML model performs better. The highest R^2 test and lowest MSE were achieved at the sixth iteration of the reduced data mutation. Thus in the case of ASOs data, collecting a large training dataset does not necessarily improve the performance of the ML model. Regarding the evaluation metrics, MSE is considered as a sensitive metric since small data mutation significantly changed the MSE behavior. Therefore, we recommend monitoring the MSE metric when testing the resilience of ML models to data mutations.

For generalization purposes, the FIUL-Data framework could apply to any ML system where the researcher has pre-trained models and can define data mutators. The flexibility of the proposed framework comes from the ability to customize multiple steps depending on the application under study. The FIUL-Data framework can be used in many interesting applications, such as studying how a trained model responds to different kinds of data faults, quantifying and evaluating the trade-off between model resilience and prediction accuracy, and investigating tuning models based on the type of fault in the data.

6   Threats to Validity

External Validity

In any research, generalization of results is important to contribute to the field of study. Despite the application of FIUL-Data on a use case from the analytical chemistry field, the general and flexible design of the proposed framework allows its application in many domains. At every stage of the framework, the user could customize the steps to suit the use case. For example, the data mutators designed in this paper could be modified as the user sees convenient. The ML models applied were trained and optimized for these ASOs datasets; in other applications, other ML models could be studied and compared.

Reproducibility of Results

To ensure the reproducibility of the results, we provide a detailed description of the methodology and the experimental setup. The controlled random split of the train and test sets supports reproducibility.

Selection of Datasets

The evaluation results of the FIUL-Data framework depend on the datasets used. The datasets are generated based on specific and controlled experiments conducted in the Chemistry Department of Karlstad University. Therefore, the ASO compounds are limited to the sequences purchased for the purpose of the experiment. This kind of data has a special characteristic: up to 3 compounds could be derivatives from the same original compound sequence. Therefore, an

ASO compound and the derivative compounds resulting from the separation process share similar characteristics, such as phosphorothioation and the frequency of nucleotide bases in the sequence. Since these characteristics are transformed into features, the underlying similarity could impact the models' performance during the mutation process despite the data's shuffling and random split before the ML application.

References

1. Chapter 16 quantitative analysis by gas chromatography sources of errors, accuracy and precision of chromatographic measurements. In: Guiochon, G., Guillemin, C.L. (eds.) For Laboratory Analyses and On-Line Process Control, Journal of Chromatography Library, vol. 42, pp. 661–687. Elsevier (1988). https://doi.org/https://doi.org/10.1016/S0301-4770(08)70088-5

2. Breck, E., Cai, S., Nielsen, E., Salib, M., Sculley, D.: The ml test score: A rubric for ml production readiness and technical debt reduction. In: 2017 IEEE International Conference on Big Data (Big Data). pp. 1123–1132 (2017). https://doi.org/10.1109/BigData.2017.8258038

3. D'Archivio, A.: Artificial neural network prediction of retention of amino acids in reversed-phase hplc under application of linear organic modifier gradients and/or ph gradients. Molecules **24**(3) (02 2019). https://doi.org/10.3390/molecules24030632, https://www.mdpi.com/1420-3049/24/3/632

4. Enmark, M., Häggström, J., Samuelsson, J., Fornstedt, T.: Building machine-learning-based models for retention time and resolution predictions in ion pair chromatography of oligonucleotides. Journal of Chromatography A **1671**, 462999 (2022). https://doi.org/https://doi.org/10.1016/j.chroma.2022.462999

5. Fornstedt, T., Forssén, P., Westerlund, D.: Basic hplc theory and definitions: retention, thermodynamics, selectivity, zone spreading, kinetics, and resolution. Anal. Sep. Sci. 5 Vol. Set **2**, 1–22 (2015). https://doi.org/10.1002/9783527678129.assep001

6. Gangolli, A., Mahmoud, Q.H., Azim, A.: A systematic review of fault injection attacks on iot systems. Electronics **11**(13), 2023 (2022). https://doi.org/10.3390/electronics11132023, https://www.mdpi.com/2079-9292/11/13/2023

7. Ghiduk, A.S., Girgis, M.R., Shehata, M.H.: Higher order mutation testing: A systematic literature review. Computer Science Review **25**, 29–48 (2017). https://doi.org/10.1016/j.cosrev.2017.06.001

8. Hellier, E., Edworthy, J., Lee, A.: An analysis of human error in the analytical measurement task in chemistry. International Journal of Cognitive Ergonomics **5**(4), 445–458 (2001). https://doi.org/10.1207/S15327566IJCE0504_5

9. Jha, S., Banerjee, S.S., Cyriac, J., Kalbarczyk, Z.T., Iyer, R.K.: Avfi: Fault injection for autonomous vehicles. In: 2018 48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W). pp. 55–56. IEEE Computer Society (2018). https://doi.org/10.1109/DSN-W.2018.00027

10. Kaiser, R.E.: Errors in chromatography. Chromatographia **4**, 479–490 (1971). https://doi.org/10.1007/BF02268820

11. Katzir, Z., Elovici, Y.: Quantifying the resilience of machine learning classifiers used for cyber security. Expert Systems with Applications **92**, 419–429 (2018). https://doi.org/10.1016/j.eswa.2017.09.053

12. Kohlbacher, O., Quinten, S., Strum, M., Mayr, B.M., Huber, C.G.: Structure–activity relationships in chromatography: Retention prediction of oligonucleotides with support vector regression. Angewandte Chemie International Edition **45**(42), 7009–7012 (2006), https://api.semanticscholar.org/CorpusID:33345638

13. Korany, M.A., Mahgoub, H., Fahmy, O.T., Maher, H.M.: Application of artificial neural networks for response surface modelling in hplc method development. Journal of Advanced Research **3**(1), 53–63 (2012)

14. Kuselman, I., Kardash, E., Bashkans ky, E., Pennecchi, F., Ellison, S., Ginsbury, K., Epstein, M., Fajgelj, A., Karpov, Y.: House-of-security approach to measurement in analytical chemistry: Quantification of human error using expert judgments. Accreditation and Quality Assurance **18**, 459–467 (2013). https://doi.org/10.1007/s00769-013-1020-9

15. Kuselman, I., Pennecchi, F., Fajgelj, A., Karpov, Y.: Human errors and reliability of test results in analytical chemistry. Accreditation and Quality Assurance **18**, 3–9 (2013). https://doi.org/10.1007/s00769-012-0934-y

16. Lotfi, R., Gholamrezaei, A., Kadłubek, M., Afshar, M., Ali, S.S., Kheiri, K.: A robust and resilience machine learning for forecasting agri-food production. Scientific Reports **12**(1), 21787 (2022). https://doi.org/10.1038/s41598-022-26449-8

17. Lu, Y., Sun, W., Sun, M.: Towards mutation testing of reinforcement learning systems. Journal of Systems Architecture **131**, 102701 (2022). https://doi.org/10.1007/978-3-030-91265-9_8

18. Ma, L., Zhang, F., Sun, J., Xue, M., Li, B., Juefei-Xu, F., Xie, C., Li, L., Liu, Y., Zhao, J., Wang, Y.: Deepmutation: Mutation testing of deep learning systems. In: 2018 IEEE 29th International Symposium on Software Reliability Engineering (ISSRE). pp. 100–111. IEEE Computer Society, Los Alamitos, CA, USA (oct 2018). https://doi.org/10.48550/arXiv.1805.05206

19. Narayanan, N., Pattabiraman, K.: Tf-dm: Tool for studying ml model resilience to data faults. In: 2021 IEEE/ACM Third International Workshop on Deep Learning for Testing and Testing for Deep Learning (DeepTest). pp. 25–28. IEEE Computer Society, Los Alamitos, CA, USA (jun 2021). https://doi.org/10.1109/DeepTest52559.2021.00010

20. Nurminen, J.K., Halvari, T., Harviainen, J., Mylläri, J., Röyskö, A., Silvennoinen, J., Mikkonen, T.: Software framework for data fault injection to test machine learning systems. In: 2019 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW). pp. 294–299 (2019). https://doi.org/10.1109/ISSREW.2019.00087

21. Papadakis, M., Kintis, M., Zhang, J., Jia, Y., Traon, Y.L., Harman, M.: Chapter six - mutation testing advances: An analysis and survey. In: Memon, A.M. (ed.)

Advances in Computers, Advances in Computers, vol. 112, pp. 275–378. Elsevier (2019). https://doi.org/10.1016/bs.adcom.2018.03.015

22. Petritis, K., Kangas, L.J., Ferguson, P.L., Anderson, G.A., Pasa-Tolić, L., Lipton, M.S., Auberry, K.J., Strittmatter, E.F., Shen, Y., Zhao, R., Smith, R.D.: Use of artificial neural networks for the accurate prediction of peptide liquid chromatography elution times in proteome analyses. Analytical Chemistry **75**(5), 1039–1048 (2003). https://doi.org/10.1021/ac0205154

23. Riccio, V., Jahangirova, G., Stocco, A., Humbatova, N., Weiss, M., Tonella, P.: Testing machine learning based systems: a systematic mapping. Empirical Software Engineering **25**(6), 5193–5254 (2020). https://doi.org/10.1007/s10664-020-09881-0

24. Risum, A.B., Bro, R.: Using deep learning to evaluate peaks in chromatographic data. Talanta **204**, 255–260 (2019). https://doi.org/10.1016/j.talanta.2019.05.053

25. Sturm, M., Quinten, S., Huber, C.G., Kohlbacher, O.: A statistical learning approach to the modeling of chromatographic retention of oligonucleotides incorporating sequence and secondary structure data. Nucleic acids research **35**(12), 4195–4202 (2007). https://doi.org/10.1093/nar/gkm338

26. Tambon, F., Khomh, F., Antoniol, G.: A probabilistic framework for mutation testing in deep neural networks. Inf. Softw. Technol. **155**(C), 107129 (mar 2023). https://doi.org/10.1016/j.infsof.2022.107129

27. Tran, A., Hyne, R., Pablo, F., Day, W., Doble, P.: Optimisation of the separation of herbicides by linear gradient high performance liquid chromatography utilising artificial neural networks. Talanta **71**(3), 1268–1275 (03 2007). https://doi.org/10.1016/j.talanta.2006.06.031

28. Vairo, T., Pettinato, M., Reverberi, A.P., Milazzo, M.F., Fabiano, B.: An approach towards the implementation of a reliable resilience model based on machine learning. Process Safety and Environmental Protection **172**, 632–641 (2023). https://doi.org/https://doi.org/10.1016/j.psep.2023.02.058

29. Webb, R., Doble, P., Dawson, M.: Optimisation of hplc gradient separations using artificial neural networks (anns): Application to benzodiazepines in post-mortem samples. Journal of Chromatography B **877**(7), 615–620 (2009). https://doi.org/10.1016/j.jchromb.2009.01.012

30. Zhang, J.M., Harman, M., Ma, L., Liu, Y.: Machine learning testing: Survey, landscapes and horizons. IEEE Transactions on Software Engineering **48**(1), 1–36 (2022). https://doi.org/10.1109/TSE.2019.2962027

31. Zheng, A., Casari, A.: Feature Engineering for Machine Learning. O'Reilly Media, Inc. (2018)

32. Zhu, Q., Panichella, A., Zaidman, A.: A systematic literature review of how mutation testing supports quality assurance processes. Software Testing, Verification and Reliability **28**(6), e1675 (2018). https://doi.org/10.1002/stvr.1675