Explainable Artificial Intelligence Using Forward-Forward Networks: A Study Involving Quantitative Analysis

Vitor L. Fabris[1], Juliane R. de Oliveira[1], Camille H. B. Silva[1], Vanessa Cassenote[1], José V. N. A. da Silva[1], Rodrigo R. Arrais[1], Renata De Paris[1]

[1]Eldorado Institute of Research – Av. Alan Turing, 275, Cidade Universitária 13.083-898 – Campinas – SP – Brazil

Abstract. *The field of eXplainable Artificial Intelligence (XAI) aims to understand the output of machine learning algorithms. We observed that the literature faults in proposing the systematic evaluation of XAI metrics and requires human perception to evaluate. This paper assesses XAI methods using the Forward-Forward (FF) algorithm from Geoffrey Hinton's proposal. Through a quantitative and critical analysis of XAI algorithms - mainly SHAP, LIME, and Grad-CAM - this study assesses the effectiveness of LIME by comparing ground truth image and LIME mask output using traditional evaluation metrics. Our contributions to this paper are to improve our understanding of the FF output using XAI and to provide a systematic strategy for evaluating XAI metrics. We demonstrate that the proposed metrics effectively highlight the features considered by the FF network when correctly or incorrectly classifying images, allowing for quantitative distinction.*

1. Introduction

Machine Learning (ML) systems have been increasingly integrated into society and used to take important decisions in sensitive areas, such as healthcare, educational assessment, credit granting, employment, criminal justice, defense, autonomous vehicles [Li et al. 2023]. However, periodic reports highlight flaws in some systems that undermine their trustworthiness [Alzubaidi et al. 2023]. For example, a system that predicts the chance of recurrence of crimes having a clear bias against black people, among other cases that have become famous.

The study of Trustworthy Artificial Intelligence (TAI) [Li et al. 2023] is of paramount importance in the ever-expanding landscape of Artificial Intelligence (AI) research. As ML models become integral to a wide range of applications, ensuring their trustworthiness is a critical imperative. TAI focuses on the need for ethical, transparent, interpretable, and accountable AI systems, fostering confidence among users and stakeholders. As AI continues to permeate various aspects of society, the study of TAI is essential for developing robust and reliable systems that align with ethical standards and societal values.

Also known as interpretability, explainability is the principle which states that the decisions and classifications of AI systems should be interpretable by its end-users, developers and stakeholders [Li et al. 2023]. This concern has given rise to an entire area of research called eXplainable AI (XAI) [Saeed and Omlin 2023], which aims to develop algorithms capable of explaining the decisions of various ML models [da Silva et al. 2023,

Holzinger et al. 2022, Das and Rad 2020]. In the context of image classification, typically these algorithms provide explanations in the form of region-of-interest visualizations, allowing interested parties to understand which parts of the image are most impactful to a model's final decision, relying on users to interpret the generated explanations visually [Ribeiro et al. 2016, Lundberg and Lee 2017, Selvaraju et al. 2017]. However, XAI frameworks lack a quantitative metric for evaluation, making it difficult to objectively measure their effectiveness. This highlights the need for developing quantitative methods to complement human-based assessments and ensure more rigorous validation of XAI systems.

In this study, we address two main contributions to the field of XAI. First, we evaluate the results of XAI techniques - SHAP, LIME and Grad-CAM - when adapted to the Forward-Forward (FF) network [Hinton 2022], a recently learning method proposed by Geoffrey Hinton to replace Backpropagation (BP). Second, we introduce new evaluation metrics to address the existing gap in XAI frameworks. These metrics assess the effectiveness of LIME by comparing ground truth images with LIME-generated masks using traditional machine learning evaluation criteria.

The following sections of this paper are structured to provide a comprehensive overview of our research. In Related Works, we review the existing literature and highlight the gaps that our study aims to address. The Methodology section details the approach, tools, and techniques employed in our experiments. Next, in Results, we present the outcomes of our experiments, followed by a thorough Discussion, where we interpret the findings and highlighting the contributions and limitations of our approach. Finally, the Conclusion summarizes the key contributions of our study and suggests potential directions for future research.

2. Related Works

Despite the growing importance of explainability in AI, research on evaluation methods for XAI algorithms has been limited, with few robust academic tools available for comprehensive analysis, as discussed in papers like [Saeed and Omlin 2023] and [Zhou et al. 2021]. Among the existing XAI methods, LIME (Local Interpretable Model-agnostic Explanations) [Ribeiro et al. 2016], SHAP (SHapley Additive exPlanations) [Lundberg and Lee 2017] and Grad-CAM (Gradient-weighted Class Activation Mapping) [Selvaraju et al. 2017] stand out as widely used tools to explain decisions of ML models.

The paper [Nguyen et al. 2021] analyzes the effectiveness of LIME, SHAP and CAM in providing clear and comprehensible explanations for image classifications, without interfering in the XAI model or computation time. The authors propose a new method: Determining the Highest-Impact Segments (DHIS). To calculate the DHIS, the authors used the intersection of the CAM heatmap and the original image segmentation from k-means algorithm. They also introduce the Intersection Over Union (IOU) metric, quantifying the accuracy of XAI methods by measuring the overlap between bounding boxes generated by the methods (LIME, SHAP, CAM) and a human-defined ground truth. Both metrics improve comparability between LIME, SHAP, and CAM, showing that LIME aligns most accurately with human interpretations, while CAM is computationally faster but less precise. DHIS enhances CAM's segmentation and highlights trade-offs between

accuracy and efficiency across XAI methods.

The study [Bitton et al. 2022] focuses on enhancing the interpretability of explanations generated by SHAP. The research highlights that while methods like SHAP are effective in explaining complex models, the generated explanations are not always directly interpretable by humans. To address this, Latent SHAP is proposed as a feature attribution framework that does not require a fully invertible transformation function, making the explanations more accessible and useful. Similarly, GradCAM is evaluated in the context of image classification, demonstrating how gradient-weighted activation maps can provide visual insights into the image regions that most influence the model's decision. Albeit not numerous, these cited studies represent some attempts to evaluate XAI methods.

Although [Nguyen et al. 2021] and [Bitton et al. 2022] evaluate XAI methods (LIME, SHAP, CAM) broadly in image classification, focusing on accuracy and computational efficiency, our study applies XAI techniques to the novel Forward-Forward neural network architecture [Hinton 2022], employing statistical metrics to quantify the accuracy of LIME explanations. To the best of our knowdged, no other previous studies in the literature address XAI methods for Forward-Forward networks, highlighting the uniqueness of our approach in tackling explainability within this novel architecture.

3. Methodology

We aim to apply three well-known XAI algorithms to evaluate whether a FF network can withstand the test of explainability: LIME [Ribeiro et al. 2016], SHAP [Lundberg and Lee 2017] and Grad-CAM [Selvaraju et al. 2017]. We select these algorithms due to their broad applicability, robustness, and reliability. Additionally, we propose three methods to quantitatively analyze the results of LIME: the first based on pixel-comparison between the original binarized image[1] and LIME's explanation; the second based on the Adjusted Mutual Information (AMI) [Vinh et al. 2010] for similar-image clustering; and the third based on the Intersection Over Union (IOU), also known as Jaccard Index. We evaluated all experiments on the MNIST dataset [Lecun et al. 1998] for its simplicity and because it was the main dataset used in [Hinton 2022]. MNIST consists of images with 28x28 pixels of handwritten digits in grayscale, with 60,000 examples for training and 10,000 examples for testing.

3.1. Forward-Forward architecture

The FF training algorithm, initially developed and studied by Geoffrey Hinton [Hinton 2022], is a training procedure to be applied to neural networks for weight updating and task convergence. It is meant to be a more biologically-plausible approach to training neural networks than the usual backpropagation (BP) algorithm, since, as stated in [Hinton 2022], the latter remains an implausible model for the learning process of biological neurons.

While BP makes an entire forward pass in the network to gather activations in order to update every layer's weights in respect to the preceding one (and the final error of that classification), the FF algorithm prioritizes updating each layer individually, "independently" of the next, until convergence or reaching maximum epochs. It is still in its

[1] A binarized image is a segmented image consisting of only 1s and 0s.

early stages of study and therefore it does not surpass BP in terms of results or training time. However, it has a lot of potential concerning memory efficiency, especially with deep networks, because it does not need to store intermediate gradients to make a full update.

Our implementation of the FF model had 2 layers of dimensions $(784, 500)$ and $(500, 500)$ respectively. It ran for a total of 250 global epochs, with each layer having an individual run of 50 epochs[2]. We used Adam [Kingma and Ba 2015] optimizer with a 0.03 learning rate, Mean Squared Error loss and Glorot Uniform initialization for the weights. As for the internal FF goodness threshold, we used a value of 1.5.

3.2. LIME

The LIME (Local Interpretable Model-Agnostic Explanations) [Ribeiro et al. 2016] algorithm is a fundamental tool in the context of ML model interpretability. Developed as a Python library, LIME is model-agnostic and improves transparency, regardless of its complexity, providing transparent and interpretable insights into the underlying reasons behind the model's decisions. Its importance is notable in approaching complex and opaque AI models as it provides understandable local explanations for humans. It is important to note that LIME can be used to explain models trained on many different data domains, but we are focusing solely on image classifiers.

LIME follows a series of steps in order to explain important image regions for a classification. First, it segments the image according to some predetermined segmentation function, separating n various mutually exclusive regions of image. Each of these regions are going to be perturbed k times following some perturbance regimen, resulting in $n \times k$ new images[3]. Afterwards, the original model being explained is going to infer some classification for each new region-perturbed image, and this classification is used by LIME to determine the influence of that region: if, because of that perturbed region, the model misses the classification by a long shot, then that region positively influences the model; if, on the contrary, the model has a more confident and correct classification, then that image negatively influences the model; finally, if the model does not change its confidence nor classification, then the region barely impacts. Visually, it is possible to combine LIME's outputs with the original image in order to highlight the important regions that influenced the model in arriving at such classification.

3.3. SHAP

The SHapley Additive exPlanation framework, liberally abbreviated as SHAP, is a collection of different explainability algorithms suited for different data modalities and ML models, all sharing a simple-to-use interface [Lundberg and Lee 2017]. It is possible to use Tree Classifiers, CNNs, MLPs, and a variety of other models. Its name is due to the primary method used to derive explanations - the Shapley value -, but many other algorithms are also implemented in this framework. For image data, it generates a discrete, coarse-grained heatmap of feature importance for a classification. While the Shapley value is the primary algorithm in this framework, other variations remain to be explored. Specifically, SHAP is divided into interfaces that work with individual model architectures. Specifically, in this work we utilize two of them: the Shapley value calculation

[2]For each global epoch, 50 individual epochs are run for each layer.

[3]n depends on the given segmentation function, while k is a hyperparameter in LIME's implementation.

and SHAP's *GradientExplainer* feature, which implements the Expected Gradients algorithm [Erion et al. 2021].

Adapted from the mathematical field of game theory, the Shapley value computes the individual impact of certain features in relation to the entire data. In summary, we assume that the data is divided into N feature, as is typically the case with tabular and image data. Using an additive approach - starting with an empty set of features and progressively adding one feature at a time - the output of the ML model is measured at each step. Each feature's contribution to the overall classification is then properly weighted to determine its individual impact.

The Expected Gradients method works a bit differently. Suppose you wish to derive an explanation from a single image. By incrementally taking different images from your data distribution (e.g. training data), interpolating and combining them with your original image, while accumulating the gradient of the resulting combined image with the specific feature you wish to understand, the importance of such feature is calculated. Now, combining this result multiple times with multiple images taken from your distribution, multiple interpolations, and proportionally weighting each contribution, the image that is obtained acts as an explanation of the important features (e.g. pixels) for that classification.

3.4. Grad-CAM

Another highly-used explainability algorithm for Convolutional Neural Networks (CNNs), the Gradient Class Activation Mapping (Grad-CAM) algorithm [Selvaraju et al. 2017], was applied to understand its potential in explaining classifications with the FF architecture. The class activation mappings that Grad-CAM produces function like a heatmap, which evaluates areas in the image where, for a given predetermined layer, the most activations occurred. These activations, together with the principle that areas with the most activation impact the most in the final activation of the model and, therefore, in the final classification, show us regions in the image that the model (through that predetermined layer) is "focusing" more, i.e. more important regions.

This is achieved by a rather simple, two-step calculation. Let A^k be the feature map activations of the given predetermined layer A, with dimensions (i, j, k) for height, width and depth (number of feature maps) respectively, and y_c be the activation score for class c, taken directly from the final activation, y. Therefore, the equations 1 and 2 are applied in sequential order, and the result is the class activation mapping for that image, layer and classification.

$$\alpha_k^c = \frac{1}{Z} \sum_i \sum_j \frac{\partial y^c}{\partial A_{ij}^k} \qquad (1) \qquad L^c_{\text{Grad-CAM}} = ReLU\left(\sum_k \alpha_k^c A^k\right) \qquad (2)$$

3.5. Pixel-comparison evaluation

LIME works by creating a mask for the image, based on the most and least important pixels for the model's prediction for that sample. This mask represents the presence or absence of influential superpixels (contiguous sets of similar pixels), taking into account the segmentation generated for the image. Therefore, our approach consists of calculating

the similarity between the binary masking of the results generated by LIME and the original binarized image, in order to output its accuracy. We assume that LIME's influential superpixels should, within some degree, resemble the binarized image since they are the only ones containing positive information that can guide a model's decision.To evaluate quantitatively the LIME results, we compute the pixel similarity using the following metrics: F-Score, Adjusted Mutual Information (AMI) and Intersection Over Union (IOU).

Initially, the pixel-comparison evaluation was inspired by the methods of statistical analysis: precision, recall and F-score. Precision ($\mathcal{P}$) measures the proportion of pixels that were correctly classified as similar between the two images. Recall ($\mathcal{R}$) measures the proportion of pixels that are similar between the two images and that were correctly classified as similar. The F-score ($\mathcal{F}$) is a metric that combines precision and recall in a single measurement, in order to understand the general performance of the results. Therefore, considering the statistical variables TP - the amount of pixels correctly classified -, FP - the amount of pixels incorrectly classified as influential - and FN - the amount of pixels incorrectly classified as not influential -, our metrics are calculated as follows:

$$\mathcal{P} = \frac{TP}{TP + FP} \quad (3) \qquad \mathcal{R} = \frac{TP}{TP + FN} \quad (4) \qquad \mathcal{F} = \frac{2\mathcal{P}\mathcal{R}}{\mathcal{P} + \mathcal{R}} \quad (5)$$

AMI measures the similarity between two clusters through mutual information [Vinh et al. 2010]. Two clusters are similar when presenting high adjusted mutual information score, i.e. score closer to 1.0, and a score close to 0 on average means that clusters are different. IOU is the traditional evaluation metric for measuring the performance of object detection algorithms. Object detection task finds the bounding box that corresponds to the position of a specific object. This metric measures the match between bounding box of ground truth and detected object. We employed AMI and IOU due to their possible use as measurements of similarity: IOU begin a spatial-similarity measure, specially when used in the context of object-detection tasks, and AMI being a numerical-similarity measure, since it derives from cluster analysis. In essence, for both metrics, we clustered the ground-truth (original MNIST images) and compared them to the cluster of "predicted" values (the LIME masking). Therefore, similarly to F-Score metric, we assume that LIME's masks should be relatively similar to the binarized ground-truth image, and this metric can precisely measure how much that is the case.

4. Results

In this section, we describe the results obtained by using LIME, SHAP and GRADCAM frameworks to explain the classification of FF model in the MNIST dataset and the results achieved by performing a quantitative evaluation on LIME outputs. The last set of experiments served as a preliminary study to automate the evaluation of LIME and other framework results using various image similarity metris.

Figure 1 shows examples of the classification of MNIST digits by a FF network, all explained through LIME. The yellow boundaries indicate influential regions for the models classifications. We do not distinguish between positive and negative influence in this case, and therefore all yellow-bounded regions are to be interpreted as simply impactful. As can be seen, these influential regions greatly match with positive areas of the image, implying that the FF architecture was able to sustain LIME's explanation.

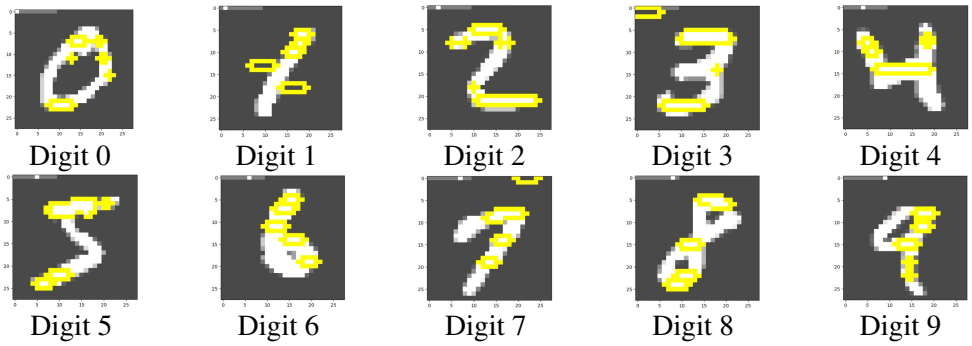


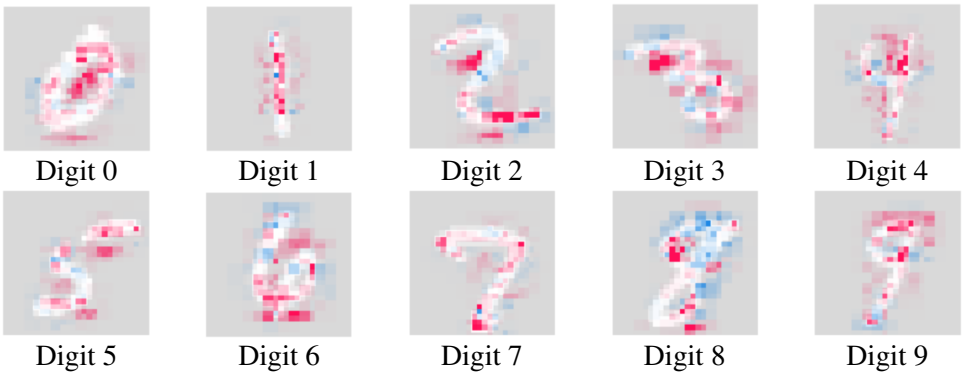
Figure 1. Examples of LIME results on FF architecture

To apply SHAP's Expected Gradients computation to a FF network, some minor changes to the architecture had to be made, specially concerning batch-handling. We exhibit both approaches with SHAP concomitantly to better indicate their similarities and differences. Figure 2 presentes the SHAP results where red markers indicate positive influence on the final classification in the mathematical sense, i.e. it contributed positively to the output of the model. In a complimentary sense, blue markers indicate negative influence. This mathematical representation of SHAP's results makes it inappropriate to interpret red markers as "good" and blue markers as "bad". The presence of both is indicative of the FF network's understanding of the digit, and one is not necessarily more valuable than the other.

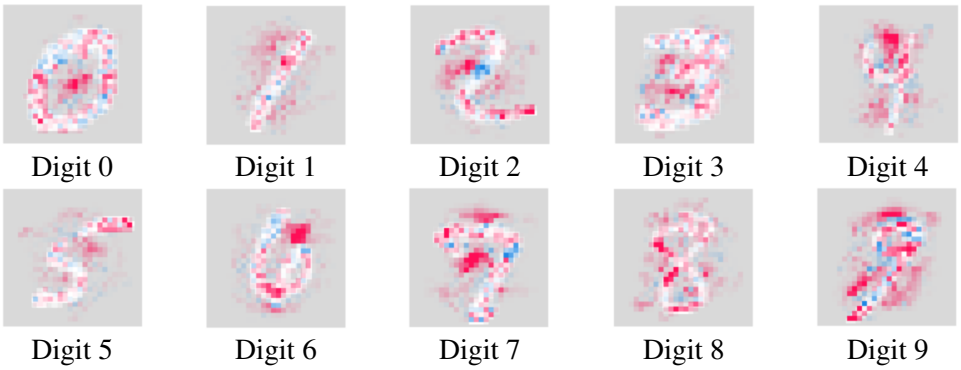
Since Grad-CAM was designed for Convolutional Neural Networks (CNNs) [Selvaraju et al. 2017], adapting it to the FF architecture proved to be a significant challenge. Our initial attempt involved modifying the Grad-CAM algorithm to work with dense layers by introducing new operations into the network's activations while trying to retain the characteristics of the original implementation. Specifically, global poolings were changed to averages, while matrix multiplications were converted to vector products. Albeit maintaining the data-transformation logic of Grad-CAM, these alterations significantly changed the algorithm's nature. The second approach was to adapt the FF architecture to be compatible with Grad-CAM by replacing its internal dense layers with convolutional layers, thereby aligning it with the explainability algorithm. Unfortunately, neither approach was successful. The results consisted of scattered pixels across images that failed to provide meaningful insights into the model, as can be seen in Figure 3. A detailed explanation of why these approaches did not work is presented in Section 5.

To evaluate the LIME results considering the pixel-comparison, we calculated the F-Score, AMI and IOU metrics 10 times for each digit from the MNIST dataset. This approach ensures that the reported metrics reflect consistent performance across multiple trials, providing a robust evaluation of the system's accuracy and reliability. The use of multiple executions and the calculation of each metric for each digit image help to mitigate the impact of outliers and variations, leading to more reliable and generalizable conclusions about the model's performance. The results shown in Figure 4 represent the average and standard deviation of 10 executions, each of which is the mean of F-score for every individual digit image.

Figures 5 and 6 show the average and standard deviation of the 10 executions of



(a) Shapley value method



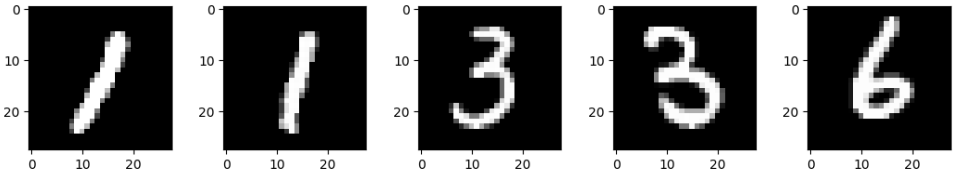
(b) Expected Gradients method

Figure 2. Results obtained with SHAP. Red indicates positive influence, while blue indicates negative influence.

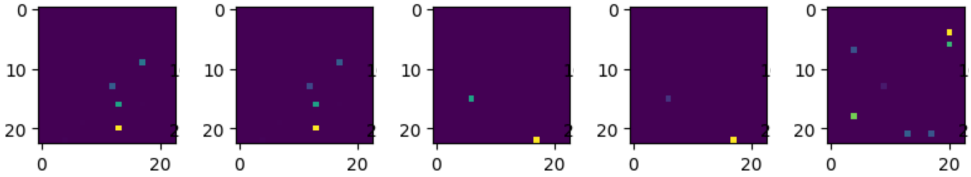
AMI and IOU metrics for each digit in the MNIST test set. An AMI score close to 1.0 indicates that the model accurately considers all image features, while a score less than 1.0 suggests that the model is considering fewer features. Similarly, for the IOU metric, a perfect match (IOU = 1) shows that the area of the ground truth object intersects perfectly with the detected object, indicating successful object detection. An IOU value less than 1 indicates an imperfect match, which can suggest incorrect detection depending on the required accuracy of the application.

5. Discussion

The application of LIME was easily adapted to the FF architecture. Due to LIME's "black-box" approach, it is compatible with any model, regardless of whether the network performs BP, is differentiable, or even if it is a ML model [Ribeiro et al. 2016]. The only requirement is that the network accepts an image as input and returns a probability vector as output. With minor adjustments to our original FF architecture implementation, we achieved this compatibility seamlessly. Furthermore, it is notable that LIME attributes influence, be it positive or negative, mostly on the positive digit information, i.e. the pixels actually comprising the digit. Our assumption concerning the pixel-wise and AMI metrics, presented in section 3.5, is based on this observed behavior from LIME's explanations.



(a) Original MNIST images



(b) Grad-CAM results

Figure 3. Examples of results obtained with Grad-CAM.

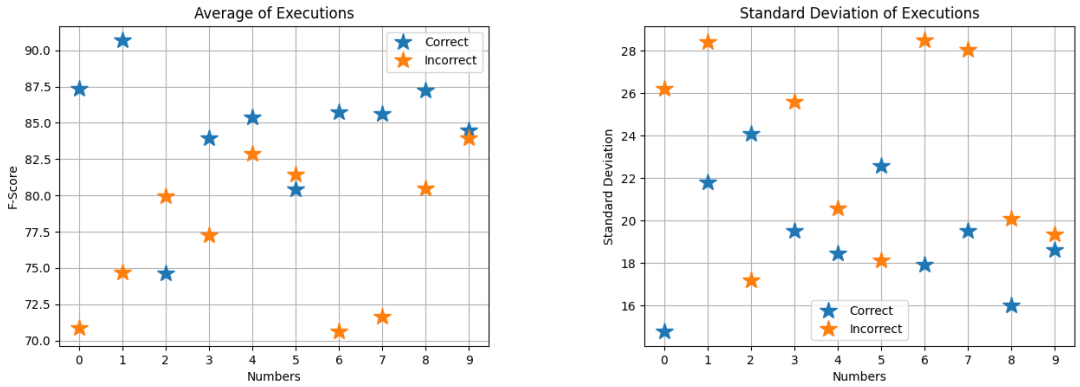


Figure 4. F-score results for each digit

It is possible to observe that SHAP does not focus exactly on the digit shapes or their general topology, but rather on the important features that define each digit's shape. For example, in Figure 2(a), SHAP identifies the most important regions of the digit 3 as the crevices on the left and right, which define its shape. Similarly, for the digit 0, it highlights the central hole; for the digit 2, the middle crevice; and for the digit 5, the two opposite crevices. The presence of red markers in these crevices - which are, by definition, absences of positive digit information - informs us that the lack of pixels in there is positively influencing our model in classifying that image as a representative of 5. This behavior of SHAP implies our previous observation, namely that it focuses more on specific topological features of each digit that, without such features, the digit would not be the same. This pattern is observed across many other digits, as can be seen in Figure 2(a). The second algorithm we experimented with was SHAP's implementation of Expected Gradients, designed specifically for use with differentiable models, i.e. models capable of gradient calculation. The explanations derived follow the same patterns as SHAP's Shapley Value calculation, attributing importance to specific topological contexts of the digits, such as the crevices in digit 3 (see Figure 2(b)). Moreover, we considered Expected Gradients better than the Shapley Value calculation, at least in terms of visualization and

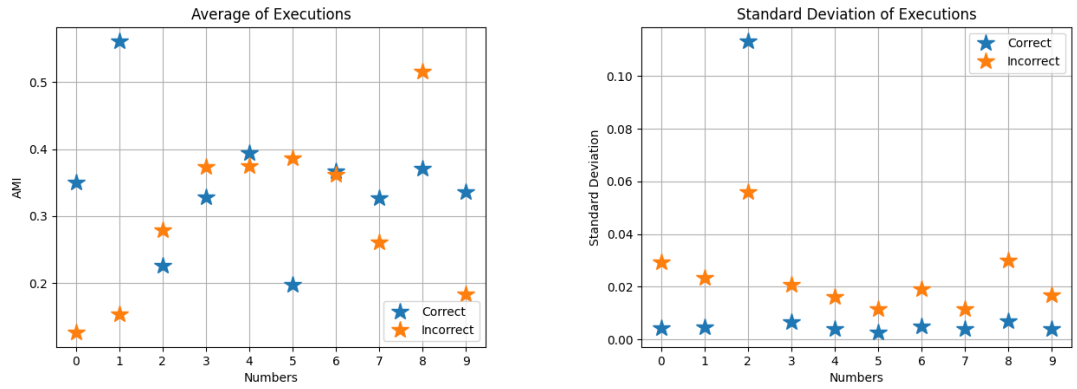


Figure 5. AMI results for each digit

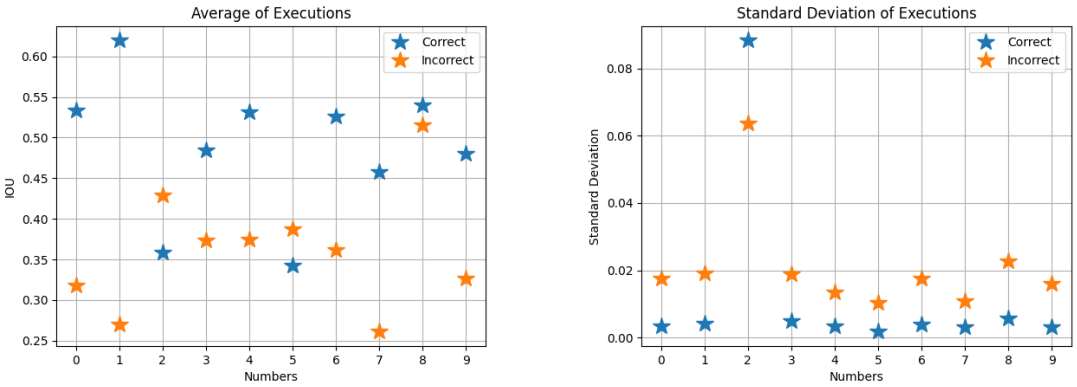


Figure 6. IOU results for each digit

understanding, because it provides more fine-grained markers.

Regarding Grad-CAM, considering that it was developed with CNNs in mind, and since the FF architecture consists solely of fully-connected layers and lacks convolutional ones, it was no surprise that Grad-CAM did not work. This prompted us to experiment with adapting the FF architecture to include convolutional layers. Our motivation came from the fact that the FF implementation we used internally depended on regular dense layers that could apply BP, but did not do so manually. We speculated that regular convolutional layers might also be applied without the BP step.

Indeed, Hinton's [Hinton 2022] warning that the FF architecture is not viable with layers that have weight-sharing was empirically confirmed. However, it is possible to design convolutional layers without weight-sharing, as demonstrated in Keras' implementation of Locally-connected convolutional layers. Nonetheless, when we attempted this experiment, the network failed to train properly, achieving accuracies around the 20% mark, which indicated that more time and resources were needed to conduct this experiment effectively.

As we expected, the F-score metric showed positive results for most of the digits. Higher F-score averages indicate a greater similarity between the LIME mask and the ground-truth (original MNIST image) pixels. Conversely, lower F-score averages indicate

less similarity between the LIME mask and the ground-truth pixels, and such imagens were usually incorrectly classified. It is important to note that the standard deviation for incorrectly classified images is higher than for correctly classified ones. This is due to the variations in the masks' topologies generated by LIME. Similar to the IOU metric, digits 2 and 5 showed low F-score averages for correctly classified images. This suggests that while this quantitative evaluation would benefit from further refinement and research, it is still capable of differentiating between correct and incorrectly classified examples.

The IOU and AMI metrics showed high performance for correct classifications and low IOU for incorrect classification. The standard deviation for AMI indicates low variability, around 0.02, except for digit 2, which shows a higher variability of about 0.12. Correct classifications have high IOU and AMI scores, indicating that the FF network considers several features of the digits used.

6. Conclusion

In this work, we presented experiments concerning the application of traditional XAI algorithms, such as LIME, SHAP and Grad-CAM, applied to the novel FF architecture proposed in [Hinton 2022]. We demonstrated successful outcomes for the first two algorithms and provided an explanation for the failure of Grad-CAM. Additionally, we evaluated three quantitative metrics for assessing XAI results, demonstrating their utility in understanding model predictions. Each experiment was discussed in detail, highlighting both the successes and limitations encountered.

For future work, we plan to refine the quantitative evaluation methods for XAI algorithms, incorporating advanced statistical analyses to enhance their effectiveness in explaining large sets of XAI outputs. Furthermore, a more in-depth investigation into the FF architecture's challanges with convolutional layers will be undertaken.

References

Alzubaidi, L., Al-Sabaawi, A., Bai, J., Dukhan, A., Alkenani, A. H., Al-Asadi, A., Alwzwazy, H. A., Manoufali, M., Fadhel, M. A., Albahri, A., et al. (2023). Towards risk-free trustworthy artificial intelligence: Significance and requirements. *International Journal of Intelligent Systems*, 2023(1):4459198.

Bitton, R., Malach, A., Meiseles, A., Momiyama, S., Araki, T., Furukawa, J., Elovici, Y., and Shabtai, A. (2022). Latent SHAP: Toward Practical Human-Interpretable Explanations.

da Silva, M. V. S., Arrais, R. R., da Silva, J. V. S., Tânios, F. S., Chinelatto, M. A., Pereira, N. B., Paris, R. D., Domingos, L. C. F., Villaça, R. D., Fabris, V. L., da Silva, N. R. B., de Faria, A. C. A. M., da Silva, J. V. N. A., de Oliveira Marucci, F. C. Q., de Souza Neto, F. A., Silva, D. X., Kondo, V. Y., and dos Santos, C. F. G. (2023). eXplainable Artificial Intelligence on Medical Images: A Survey.

Das, A. and Rad, P. (2020). Opportunities and Challenges in Explainable Artificial Intelligence (XAI): A survey. *CoRR*, abs/2006.11371.

Erion, G., Janizek, J. D., Sturmfels, P., Lundberg, S. M., and Lee, S.-I. (2021). Improving performance of deep learning models with axiomatic attribution priors and expected gradients. *Nature Machine Intelligence*, 3(7):620–631.

Hinton, G. (2022). The Forward-Forward Algorithm: Some preliminary investigations.

Holzinger, A., Saranti, A., Molnar, C., Biecek, P., and Samek, W. (2022). *Explainable AI Methods - A Brief Overview*, pages 13–38. Springer International Publishing, Cham.

Kingma, D. P. and Ba, J. (2015). Adam: A method for stochastic optimization. In Bengio, Y. and LeCun, Y., editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Lecun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

Li, B., Qi, P., Liu, B., Di, S., Liu, J., Pei, J., Yi, J., and Zhou, B. (2023). Trustworthy AI: From Principles to Practices. *ACM Comput. Surv.*, 55(9).

Lundberg, S. M. and Lee, S.-I. (2017). A unified approach to interpreting model predictions. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 4768–4777, Red Hook, NY, USA. Curran Associates Inc.

Nguyen, H. T. T., Cao, H. Q., Nguyen, K. V. T., and Pham, N. D. K. (2021). Evaluation of explainable artificial intelligence: SHAP, LIME, and CAM. In *Proceedings of the FPT AI Conference*, pages 1–6.

Ribeiro, M. T., Singh, S., and Guestrin, C. (2016). "Why Should I Trust You?": Explaining the Predictions of Any Classifier.

Saeed, W. and Omlin, C. (2023). Explainable AI (XAI): A systematic meta-survey of current challenges and future opportunities. *Knowledge-Based Systems*, 263:110273.

Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., and Batra, D. (2017). Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 618–626.

Vinh, N. X., Epps, J., and Bailey, J. (2010). Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance. *Journal of Machine Learning Research*, 11(95):2837–2854.

Zhou, J., Gandomi, A. H., Chen, F., and Holzinger, A. (2021). Evaluating the Quality of Machine Learning Explanations: A Survey on Methods and Metrics. *Electronics*, 10(5).