Vertical Split Learning-based Identification and Explainable Deep Learning-based Localization of Failures in Multi-Domain NFV Systems

Fatima Ezzeddine[1,2], Omran Ayoub[1], Davide Andreoletti[1], Massimo Tornatore[3], and Silvia Giordano[1]

[1]Scuola Universitaria Professionale della Svizzera italiana, Lugano, Switzerland, Email: {name.surname}@supsi.ch
[2]Università della Svizzera italiana, Lugano, Switzerland, Email: {name.surname}@usi.ch
[3]Politecnico di Milano, Milan, Italy, Email: {name.surname}@polimi.it

***Abstract*—Automated failure management in Network Function Virtualization (NFV) systems continues to gain significant attention as it allows identifying and mitigating failures in a timely manner, ensuring continuous and stable operation of services. In multi-domain systems, where services are provisioned across multiple domains, each domain is managed by a unique single-domain orchestrator (SDO), the problem of automated NFV failure management takes another dimension as it requires a privacy-preserving collaboration among the SDOs. This is due to the fact that SDOs are not willing to share private and business-critical information of their network to different parties. In this paper, we focus on the problem of failure identification and localization in NFV systems in multi-domain networks where SDOs collaborate, in a distributed privacy-preserving learning scheme, to train a single neural network without sharing any raw data. To this end, we propose a Vertical Split Learning (VSL)-based approach with a client-server architecture for failure identification and localization over vertically partitioned data. Additionally, we utilize Explainable Deep Learning (XDL) frameworks, namely Integrated Gradients and DeepLIFT, on the failure identification server model to locate the failures without accessing the original data or features and without training a separate localization model. We compare our approach to centralized baseline approaches, and illustrative numerical results show that our proposed solution preserves a performance close to the one achievable with a centralized approach and localizes failures with an accuracy of 83% without the necessity of training a new localization model.**

***Index Terms*—Network function virtualization, failure management, multi-domain networks.**

I. Introduction

Network Function Virtualization (NFV) plays a crucial role in supporting 5G services that require high reliability as it enables the implementation of network functions as virtualized entities, which can be dynamically allocated and managed according to the needs of different 5G/6G services [1], [2]. This flexibility is essential for network operators to deploy new network functions and services to support the evolving requirements of 5G use cases such as manufacturing, healthcare, and autonomous driving.

Recently, many studies in NFV have investigated the end-to-end management and orchestration of network resources [3], [4] while other studies have focused on applying machine learning-based techniques to automate network management [5]–[7], with the final aim of guaranteeing a reliable service provisioning and reducing service downtime. In particular, automated failure management has attracted a lot of attention, as failures of virtual network functions (VNFs) (which are generally more likely than those in hardware-based solutions [8]) can result in cascaded service outages and disruptions if not identified and dealt with in a timely manner. The majority of the studies, however, focus on single-domain networks, i.e., network management is delimited for a single administrative domain, and therefore, propose centralized learning approaches for failure management [6], [7], [9]–[11].

Multi-domain networks (MDN) [12], [13] are networks that span across multiple administrative domains that operate independently (i.e., each domain is responsible for managing its own network resources). Fig. 1 shows a schematic representation of an MDN consisting of three domains, where a service consisting of three VNFs (one VNF is deployed in each domain), each of which performs a specific task. In these networks, the application of centralized learning algorithms is infeasible, as the data required to feed learning models is distributed among different domains which are managed by different entities, rendering the aggregation of data not possible as the data is considered sensitive and critical from a business standpoint [14]. The entities managing the various domains (i.e., the single-domain orchestrators) must partner in a privacy-preserving collaborative learning scheme, as they are not willing to disclose private and business-critical information relating to their domains. Consequently, an automated failure management system for MDNs demands a privacy-preserving collaboration among the various single-domain orchestrators. Such a system should balance between data privacy (i.e., should allow single domains to retain their data privacy when sharing data) and overall performance in identifying and localizing NFV failures.

In this work, we focus on the problem of NFV failure identification and localization in MDNs. We consider that service chains (SCs) are deployed across a MDN and that single domains have to cooperate in a privacy-preserving manner, i.e., without sharing sensitive information regarding their single domain networks. Since SCs are provisioned across multiple domains, the failure data of a SC is vertically

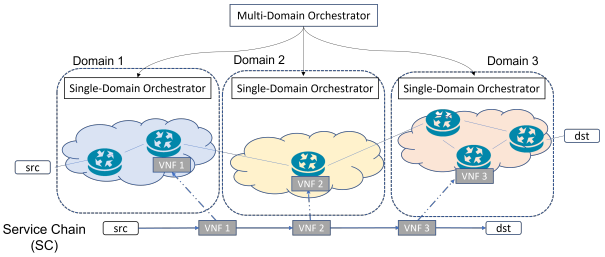

Fig. 1. A service chain (i.e., a set of chained VNFs providing a service) provisioned across a MDN.

partitioned among the single domains. In other words, each SDO has a set of features that describes the VNFs deployed within its single domain network. The parties involved are the Multi-domain Orchestrator (MDO) and a given number of SDOs managed by the MDO. We propose a privacy-preserving failure-management solution based on Vertical Split Learning (VSL) to discriminate between failure causes and failure locations (per failure location, we refer to the single domain in which the failure occurs). We also propose an approach that exploits eXplainable Deep Learning (XDL) techniques for failure localization. The proposed XDL-based approach can be used at the server side of the failure identification model and does not require the training of a separate model for failure localization like in the VSL-based approach. Numerical results show that our proposed privacy-preserving VSL-based and XDL-based approaches can achieve performance comparable to centralized models in terms of failure identification and localization. A comparison of the XDL-based approach and the VSL-based approach in terms of performance shows that the XDL-based approach can detect failure locations paying off a negligible decrease in accuracy compared to the VSL-based approach while simplifying the whole localization procedures.

The remainder of the paper is organized as follows. In Sec. II we provide a comprehensive overview of relevant literature. Sec. III provides background on VSL and the XDL frameworks applied in our work. Sec. IV describes the problem statement and the privacy requirements. Sec. V outlines our proposed approach. In Sec. VI we detail the evaluation settings. Sec. VII discusses numerical results and Sec. VIII concludes the paper.

II. Related Work

A. Multi-Domain NFV

An overview of multi-domain orchestration for NFV is presented in [15], which examines the main challenges and introduces current technologies. In [16], authors present a survey on distributed NFV multi-domain orchestration, and in [16], authors discuss the privacy aspect of multi-domain orchestration. Other works have focused on service chaining and/or automation, including orchestration and scaling, in multi-domain NFV systems [14], [17]–[19]. In [14], authors propose a lightweight, privacy-aware orchestration framework for multi-domain NFV/SDN, reducing the use of sensitive information to reduce privacy and security risks. Ref. [17] addresses the problem of allocating VNFs and Forwarding Graph (VNF-FG) to meet Quality of Service requirements while considering constraints of the underlying infrastructure such as placing a service on multiple non-cooperative domains. Ref. [18] investigated the problem of VNF autoscaling in MDNs by leveraging federated deep learning models, while [20] proposes a novel privacy-preserving reinforcement learning algorithm for multi-domain virtual network embedding that aims to protect the data of Internet Service Providers (ISPs) from a third-party entity.

B. ML-based Fault Management for NFV

Several recent studies have investigated the application of machine learning techniques for automated fault management in NFV systems [6], [7], [9]–[11]. In particular, the problem of anomaly detection has attracted the most attention [7], [9], [10]. In [7] proposes an NFV anomaly detection method based on the NoisyStudent technique and utilizes existing methods of explainable AI, namely, Shapley additive explanations (SHAP), for failure localization. In [9], authors present a data-driven approach using autoencoders based on recurrent neural networks for anomaly detection in VNFs while authors in [10] introduce a framework for unsupervised anomaly detection in a distributed environment utilizing real-time monitoring data. In [6], authors present an anomaly detection approach by monitoring SLA violation scenarios in NFV environments and incorporating root-cause localization. Ref. [11] investigates the behavior of multiple VNFs along service chains in NFV environments, and uses a regression analysis technique to detect abnormal behavior and identify causes of performance uncertainties.

In contrast to previous research, our work is the first to address the problem of automated failure identification and localization in multi-domain NFV systems. More specifically, we consider vertically partitioned data relating to NFV systems and apply client-server vertical split learning architectures to identify the type of failure and to localize the domain in which the failure occurred. Additionally, we investigate, for the first time, how the server can exploit explainable deep learning techniques to identify the location of the failure (i.e., in which domain the failure occurs), and thus reveal information about the failure location without access to the domain's local features (i.e., without clients sharing data relative to failures). While this capability permits localizing failures without the need for training a separate model, we argue that it can be also used to reveal sensitive information that the clients are not willing to disclose (in our case, the client responsible for failure).

III. Background

A. Vertical Split Learning

Split Learning (SL) is a type of federated learning approach, i.e., it allows a set of distrustful parties to jointly train a neural network-based system in a privacy-preserving manner [21], [22]. Through SL, several parties can jointly train deep neural

network model by keeping data locally, and hence, increase the privacy protection of their data. The SL approach can be either vertical (i.e., VSL), or horizontal (i.e., HSL), based on how the dataset is split among the involved parties. In VSL, data is split vertically among participants that have different features (i.e., each participant has a subset of different features or columns of a data record). VSL is well-suited for the collaboration of entities holding different sub-sets of features related to the same observation as is the case in a multi-domain virtual network.

At the architectural level, a VSL system consists of a neural network distributed among a set of L clients and a server (there are no constraints on L, it can vary based on the number of independent clients/entities that the use case has). On the clients' side, each client controls its own local part of the neural network, which is composed of a set of N layers, i.e., $L_0^{(i)}$,$L_1^{(i)}$, ..., $L_N^{(i)}$, for the generic i-th client. The last layer, i.e., $L_N^{(i)}$ is referred to as the cut layer. Note that the local networks can be different among the clients (e.g., have a different number of layers). On the server's side, the server owns the remaining layers of the global neural network, which is composed of M layers, from the cut layer until the final layer $L_{N+1}^{(i)}$, $L_{N+2}^{(i)}$, …, $L_{N+M}^{(i)}$. In the training process, each client feeds its own features to its local neural network F_i up to the cut layer, then the clients send the gradients of their networks' final layers to the server (cut layer), which merges them and continues the training process until the output layer. Various merging techniques can be used, such as concatenation, element-wise max pooling, element-wise average pooling, element-wise product, and element-wise sum. Finally, using the gradients of the server model, the clients can calculate the gradients for each batch of samples and use them to update their own models.

B. Explainable Deep Learning

Explainable AI (XAI) is a branch of artificial intelligence that aims to make AI models more transparent and understandable to humans, and its goal is to provide explanations for the decisions made by AI models. Feature attribution is an XAI approach that highlights which and how features of the input data influence the model's decision.

In this study, we apply two gradient-based feature attribution methods, *Integrated Gradients* [23] and *Deep Learning Important Features (DeepLIFT)* [24], [25]. Both approaches compute the attribution of each input feature on the final output by estimating each feature's impact on the model's decision. This contribution, referred to as the *contribution score*, is computed by measuring the impact that small changes in the input have on the output of the DL model. In Sec. V-B we explain in more detail how we can exploit these techniques, at the server side, to identify the location of failure in an NFV system in a MDN even when the involved parties (clients) do not collaborate with the server to perform failure localization.

IV. Problem Formulation

We model the problem of privacy-preserving NFV failure identification and localization in an MDN as a multi-class VSL-based classification problem with the aim of identifying the failure root-cause of an NFV system failure and its location (domain in which failure occurs). We consider five failure classes, namely, BGP (Border Gateway Protocol) hijacking, BGP injection, node down, interface down, packet-loss-delay (described in Sec. VI), and a normal (no failure) class.

We consider that each domain, corresponding to a client in the proposed VSL scheme, owns a set of features describing the behavior of an NFV system deployed across the multiple domains in the network. The developed system needs to identify the type of VNF failures and their location within the network while keeping the data and the sensitive information of the clients locally, which is taken into consideration in the proposed system, where clients share only the output of their local model (the cut layer).

We adopted VSL with L clients. These L clients aim to build a robust classifier, at the server side, without sharing data. Each client i, has a set of j features (f_{i1}, ... , f_{ij} where f_ij is the j^{th} feature of client i) that correspond to a failure record. The features include information about the network infrastructure, devices, and systems, as well as information about the users and the data and services that are impacted by a failure (detailed in Sec. VI). We consider three main entities in the multi-domain NFV system (Fig. 1):

- Multi-domain orchestrator (MDO): The MDO owns and operates the physical and virtual networks that make up the MDN. The MDO is responsible for putting together all services and managing resources across all domains. In our scheme, the MDO is also considered a client and contributes by a set of features pertaining to administration.
- Single-domain orchestrator (SDO): The SDO owns and operates the VNFs in its own domain, accesses resources, and communicates with other domains. It may also be responsible for providing information to the MDO that is necessary for the proper functioning of the network. Each domain manages a specific number of VNFs and is considered a client in the VSL architecture.
- Server: The server is a third party defined as the entity that possesses the server model in the VSL architecture. Note that the server could as well be the MDO however we consider it, without loss of generality of our proposed method, to be a separate third entity.

The privacy requirements for the entities in this system can be divided into two main categories:

- Protection of the features of each NFV domain: This requirement aims to ensure that the sensitive information about the domain and its data, such as network configuration and information, is protected from unauthorized access or disclosure.
- Protection of the location of failures: This requirement is considered in the case where SDOs do not collaborate to perform an automated failure localization framework (but only failure identification). In this context, the location of failures within the MDN is considered protected

TABLE I
TYPES OF FEATURES AND THEIR DESCRIPTION

Feature	Definition
cpu-util	CPU utilization
admin-status	Interface status
network-incoming-packet-rates	Network incoming packet rates
network-outgoing-packet-rates	Network outgoing packet rates
tx-pps	TX packet per second
rx-ppx	RX packet per second
prefix-activity-received-current-prefixes	Information on prefix activity

TABLE II
DESCRIPTION OF CLASSES AND THEIR DISTRIBUTION IN DATASETS [26]

Class	Description	Dataset 1	Dataset 2
normal	no failure	3870	3505
BGP hijacking	Hijack of origin route	95	377
BGP injection	Injection of anomaly route	191	329
node-down	unanticipated reboot of NFV	55	140
interface-down	non-functional interface	233	157
packet-loss-delay	loss or delay of packets	1525	825

information and should not be disclosed to other involved entities.

V. NFV FAILURE MANAGEMENT IN MDN

This section describes our two proposed approaches for failure identification and localization, which are shown in Fig. 2. The first approach (upper box) is solely based on VSL, and it consists of two components. The first component, *Failure Identification* (step 1 in figure), involves the training of a Vertical Split Neural Network with a specified number of clients (procedure for this is outlined in Sec. V-A). Similarly, the second component, *Failure Localization* (step 2A), trains a vertical split neural network for the same set of clients as in *Failure Identification*. The second approach (lower box) shares the same first component with the first approach (it performs *Failure Identification* via VSL) and exploits XDL techniques for failure localization. Note that in this second approach, the clients do not contribute to model training for failure localization (i.e., the clients do not collaborate to build an automated failure localization framework). The server, however, exploits XDL techniques to identify the most contributing neurons aiming to localize failures (this procedure in Sec. V-B).

A. VSL-based Failure Identification and Localization

The implementation of the VSL involves the use of L clients and 1 server. Each client has a local model that consists of three layers with 32, 48, and 64 neurons in each layer, respectively. The activation function used is the ReLU (rectified linear unit) activation function. The server model, which is the global model, also has three layers. The first layer size is equal to the number of neurons after merging the contributions from all clients, the second layer has 64 neurons, and the third layer has the number of classes that the model is intended to classify. We train the model with 25 epochs using the Adam optimizer to optimize the negative log-likelihood loss function (the number of epochs was sufficient for the model to converge). We further note that different architectures and parameters were explored during the training process to find the best configuration for the model. As for choosing the merging technique at server-side model, we conduct an experiment considering the different merging techniques of the cut layers of the involved clients, namely, *concatenation, element-wise max pooling, average pooling, summation, and product* (see Sec. III-A), and adopted the technique with best performance which was the concatenation technique in our case (discussed in more detail in Sec. VII).

B. XDL-based Failure localization

The XDL-based failure localization approach (step '2B' in Fig. 2) consists in localizing the failure by utilizing the contribution scores of the neurons of the concatenated layer of the clients (i.e., the layer concatenating the cut layer of each client) of the model used for failure identification, without the need to train a separate model specific for failure localization. More specifically, the server does not have access to the VNF's initial features and only possesses knowledge of the gradients of the cut layer, which are concatenated to form the input for the server's model.

We apply two XDL frameworks, namely, Deep LIFT and Integrated Gradients, to compute importance scores. Then, to correlate importance scores with failure localization, we denote K as the number of most influential neurons to be relied upon for the localization of features. For instance, if $K = 5$, and the top four of the five most contributing neurons originate from the cut layer of a particular client, then the failure is deemed to be localized on the client's side (i.e., in the domain corresponding to that client). In the event of a tie, where there is equality in the contribution of neurons across two or more locations, we proceed by evaluating an additional neuron until a clear differentiation is established.

VI. EVALUATION SETTINGS

A. Network Topology and Dataset

We use the "ITU AI/ML in 5G" challenge dataset [26], [27] generated within an NFV-based test environment that simulates a 5G IP core network. The topology of the NFV testbed, consisting of five Virtual Network Functions (VNFs), is depicted in Fig. 3 and consists of 5 nodes, two IP core nodes (TR-01 and TR-02), two internet gateway routers (IntGW-01 and IntGW-01), and a router reflector (RR-01), each hosted on a different Virtual Machine (VM). Performance metrics such as CPU utilization and network incoming/outgoing packet rates are collected from each VNF per minute, as listed in Table I. The dataset is comprised of two extensive collections of failure records, which have been divided into two datasets. Dataset 1, consists of 5969 records, with 3870 records with no-fault records and 2099 failure records, and dataset 2, consists of 5333 records, with 3505 normal and no-fault records and 1828 failure records). Tab. II describes the failure classes and their distribution in both datasets. For the failure location labels of dataset 1, 415 failures are located in the node TR-01, 405 failures in TR-02, 410 failures in IntGW-01, 411 failures in IntGW-02, and 172 failures in (RR-01). For the failure

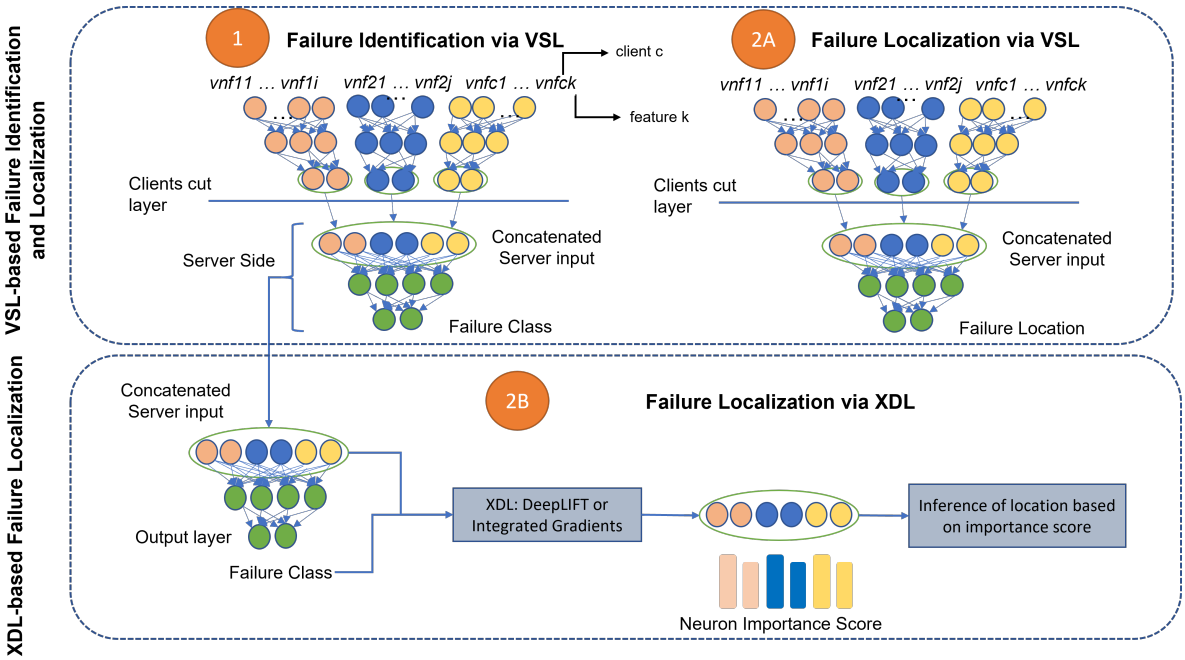

Fig. 2. Overall framework that consists of two components: (1) Failure Identification using VSL and (2) Failure Localization, which is further divided into two sub-components: (2A) Failure Localization via VSL and (2B) Failure Localization via XDL.

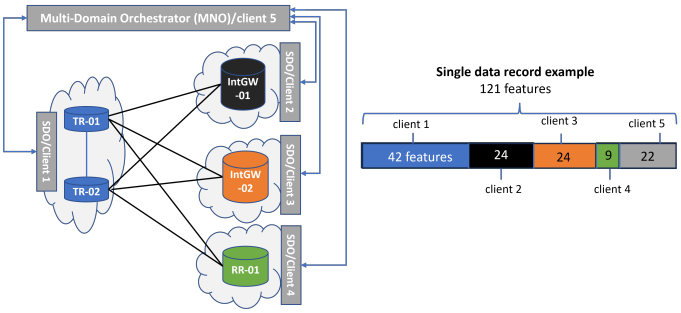

Fig. 3. NFV system and domains

location labels of dataset 2, 235 failures are located in the node TR-01, 245 failures in TR-02, 272 failures in IntGW-01, 244 failures in IntGW-02, and 127 failures in RR-01. Note that failure records that do not refer to any location are removed from the evaluation.

We divide the dataset vertically among 5 clients. We consider 4 SDOs in the network, where each domain is represented by a unique client, and the MDO, which is a separate entity representing the fifth client. Fig. 3 shows the division of the network into the four domains and the number of features corresponding to each client.

B. Benchmark Approaches

As a benchmark approach for failure identification, we consider two centralized ML approaches proposed and tested in [26], namely, multi-layer perceptron (MLP) and a graph convolutional network (GCN). We compare the performance of our proposed VSL-based learning approach to that of centralized ML approaches in terms of accuracy, precision, recall, and f1-score. For failure localization, we develop a centralized benchmark approach based on eXtreme Gradient Boosting (XGB) model to which the performance of our proposed XDL-based and VSL-based is compared to.

VII. Experimental Results

A. Failure Identification

Selection of Cut Layers' Merging Technique. We first conduct an experiment considering different merging techniques of the cut layers, namely, *concatenation, element-wise max pooling, average pooling, summation, and product* (discussed in Sec. III-A), of the involved clients. The aim of this experiment is to select the best-performing merging approach of clients' cut layers on the server side.

We use dataset 1 for this evaluation with 5-fold cross-validation. Table III reports the classification results of the various merging strategies. Results show that the concatenation technique outperforms the other strategies with an accuracy of 0.86, a precision of 0.87, and a recall of 0.86. *Sum*, *Average* and *Max* show an acceptable performance slightly lower than that of *concatenation* with an F1-score around 0.8, while *Product* is the least performing with an F1-score of 0.51. In the following experiments, we consider the *concatenation* technique for the VSL-based failure identification approach.

VSL-based vs. Centralized. We now compare the performance of our proposed approach to the centralized benchmark approaches (GCN and MLP of [26]). Tab. IV reports the classification results also showing the breakdown for each of the failure classes. The VSL-based method attains a performance slightly below the centralized approaches (accuracy of 0.86 with a difference of 0.01 and 0.05 when compared to the

TABLE III
COMPARISON OF FIVE MERGING TECHNIQUES

Merging technique	Accuracy	Precision	Recall	F1-score
Concatenation	**0.86**	**0.87**	**0.86**	**0.85**
Sum	0.80	0.81	0.80	0.79
Average	0.84	0.86	0.84	0.83
Max	0.80	0.79	0.80	0.78
Product	0.64	0.43	0.64	0.51

centralized MLP and GCN approaches, respectively). Note that the normal and packet loss/delay classes exhibit slightly lower precision than the centralized approaches, whereas the other classes surpass it. This indicates that clients in a multi-domain NFV system can maintain the privacy of their features paying off only a minor decrease of performance. Results also show that performance is satisfactory across all classes, with a precision above 0.82 and a recall ranging between 0.62 and 1.0, with the exception of BGP injection, which exhibits low recall but high precision and recall (1.0 and 0.45, respectively). Despite some differences in performance with respect to centralized approaches, we can conclude that the VSL-based approach manages to strike a balance between privacy and performance. Finally, we evaluate the effectiveness of the developed model on dataset 2. Results show that our proposed approach achieves a similar performance as that on dataset 1, with an accuracy of 0.87 and precision and recall of 0.88 and 0.87, respectively.

B. Failure Localization

We now focus on the numerical results of failure localization by comparing four different approaches:

- VSL-based: VSL-based model developed by the collaboration of all clients with the concatenation merging strategy.
- IG (XDL-based): application of IG on the VSL server-side model for failure identification.
- DeepLIFT (XDL-based): application DeepLIFT on the VSL server-side model for failure identification.
- Centralized: centralized model based on eXtreme Gradient Boosting (XGBoost).

We perform the training of the VSL-based approach and the centralized approach on dataset 1. Note that no training is required for the XDL-based approaches, DeepLIFT and IG. Testing is performed using dataset 2.

Table V reports failure localization results. The VSL-based approach shows an accuracy and F1-score of 0.91 and 0.9, respectively, comparable to that of the centralized approach (accuracy of 0.95 and F1-score of 0.94). This shows that the VSL-based approach manages to strike an acceptable balance between data privacy and performance also for failure localization. Considering the XDL-based approaches, IG achieves an accuracy of 0.83 and an F1-score of 0.86 while DeepLIFT shows a slightly lower performance with an accuracy of 0.77 and an F1-score of 0.82. The performance of the IG approach, in particular, shows that using XDL techniques can allow us to identify, with an acceptable tradeoff in accuracy (0.86 F1-score instead of 0.9 of the VSL-based approach), the failure location without requiring any additional training, which represents an advantage in terms of complexity and data overhead. These results show that XDL-based techniques can be incorporated in the developed system failure identification and localization for classification purposes while reducing complexity. The choice between XDL and VSL, however, depends on the desired outcome, with XDL emphasizing on simplicity (no specific training required) with acceptable accuracy and VSL emphasizing compensating the accuracy but requiring specific training and data overhead.

C. Discussion on Privacy Requirements

The proposed system's security in terms of privacy requirements can be described as follows:

Protection of the clients features: The proposed system employs the use of VSL for failure identification and localization to safeguard the data of individual clients (domains). The server is restricted to access only the gradients of the cut layer and does not possess knowledge or access to the parameters of the local client models. As discussed in [21], the fact that the server is unable to access the client's configuration renders it incapable of launching a reverse attack on the client's model, thereby ensuring the confidentiality of the client's features from both the server and client's side.

Protection of the failure location: The proposed VSL-based system's ability to protect the location of failures is limited. The results obtained indicate that through the use of XDL frameworks, information about the localization of failures can be accessed without clients' collaboration or consent. It is important to note that this aspect of the system requires further attention and improvement in order to ensure full protection of user privacy. If the clients provide their consent for the localization of failures, the system can leverage this information to provide added value by enabling the localization of data without the need for additional training.

VIII. CONCLUSION

We focused on the problem of failure identification and localization for NFV systems in multi-domain networks where single domains are managed by different entities, namely, the single-domain orchestrators, which are only willing to collaborate to build an automated privacy-preserving failure identification and localization system. We consider that service chains, consisting of various virtual network functions, are provisioned across the multi-domain network and that failure data pertaining to a service chain is partitioned vertically across the various domains. To this end, we propose a Vertical Split Learning (VSL)-based approach for failure identification and localization, and an Explainable Deep Learning (XDL)-based for failure localization. The VSL approach showed high accuracy in failure identification, reaching 86%, which was comparable to the accuracy of centralized models. Furthermore, we demonstrated the ability of XDL to offer a deeper understanding of the location of failures through an additional step that explains the VSL server failure identification model, reaching an accuracy of 83%.

TABLE IV
PERFORMANCE COMPARISON OF VSL VS CENTRALIZED

scheme	criteria	normal	BGP hijacking	BGP injection	Node down	Interface down	Packet loss/delay	Accuracy
VSL (ours)	precision	0.82	1.00	1.00	1.00	1.00	0.93	
	Recall	0.98	0.71	0.45	1.00	0.95	0.62	0.86
	F1-score	0.89	0.83	0.62	1.00	0.97	0.74	
GCN	precision	0.89	0.97	0.98	0.99	0.99	0.98	
	Recall	0.99	0.70	0.96	1.00	1.00	0.62	0.91
	F1-score	0.94	0.82	0.97	1.00	1.00	0.76	
MLP	precision	0.90	0.97	0.97	1.00	0.96	0.66	
	Recall	0.91	0.71	0.92	0.99	0.96	0.73	0.87
	F1-score	0.91	0.82	0.95	1.00	0.96	0.69	

TABLE V
FAILURE LOCALIZATION RESULTS

Approach	Accuracy	Precision	Recall	F1-score
VSL-based	0.91	0.91	0.90	0.90
XGBoost (Centralized)	0.94	0.95	0.94	0.94
DeepLIFT	0.77	0.91	0.77	0.82
IG	0.83	0.92	0.83	0.86

IX. ACKNOWLEDGMENT

This work was supported by the Swiss Government Excellence Scholarship (ESKAS) and the European Union under the Italian National Recovery and Resilience Plan (NRRP) of NextGenerationEU, partnership on "Telecommunications of the Future" (PE00000001 - program "RESTART").

REFERENCES

[1] D. Basu, R. Datta, and U. Ghosh, "Softwarized network function virtualization for 5g: Challenges and opportunities," *Internet of Things and Secure Smart Environments*, pp. 147–192, 2020.

[2] Z. Zhang, Y. Xiao, Z. Ma, M. Xiao, Z. Ding, X. Lei, G. K. Karagiannidis, and P. Fan, "6g wireless networks: Vision, requirements, architecture, and key technologies," *IEEE Vehicular Technology Magazine*, vol. 14, no. 3, pp. 28–41, 2019.

[3] R. Guerzoni, I. Vaishnavi, D. Perez Caparros, A. Galis, F. Tusa, P. Monti, A. Sganbelluri, G. Biczók, B. Sonkoly, L. Toka *et al.*, "Analysis of end-to-end multi-domain management and orchestration frameworks for software defined infrastructures: an architectural survey," *Transactions on Emerging Telecommunications Technologies*, vol. 28, no. 4, p. e3103, 2017.

[4] J. Baranda, J. Mangues-Bafalluy, I. Pascual, J. Nunez-Martinez, J. L. De La Cruz, R. Casellas, R. Vilalta, J. X. Salvat, and C. Turyagyenda, "Orchestration of end-to-end network services in the 5g-crosshaul multi-domain multi-technology transport network," *IEEE Communications Magazine*, vol. 56, no. 7, pp. 184–191, 2018.

[5] J. Gallego-Madrid, R. Sanchez-Iborra, P. M. Ruiz, and A. F. Skarmeta, "Machine learning-based zero-touch network and service management: A survey," *Digital Communications and Networks*, vol. 8, no. 2, pp. 105–123, 2022.

[6] J. Hong, S. Park, J.-H. Yoo, and J. W.-K. Hong, "Machine learning based sla-aware vnf anomaly detection for virtual network management," in *2020 16th International Conference on Network and Service Management (CNSM)*. IEEE, 2020, pp. 1–7.

[7] S. S. Johari, N. Shahriar, M. Tornatore, R. Boutaba, and A. Saleh, "Anomaly detection and localization in nfv systems: an unsupervised learning approach," in *NOMS 2022-2022 IEEE/IFIP Network Operations and Management Symposium*. IEEE, 2022, pp. 1–9.

[8] B. Han, V. Gopalakrishnan, G. Kathirvel, and A. Shaikh, "On the resiliency of virtual network functions," *IEEE Communications Magazine*, vol. 55, no. 7, pp. 152–157, 2017.

[9] A. Diamanti, J. M. S. Vilchez, and S. Secci, "Lstm-based radiography for anomaly detection in softwarized infrastructures," in *2020 32nd International Teletraffic Congress (ITC 32)*. IEEE, 2020, pp. 28–36.

[10] F. Schmidt, A. Gulenko, M. Wallschläger, A. Acker, V. Hennig, F. Liu, and O. Kao, "Iftm - unsupervised anomaly detection for virtualized network function services," in *2018 IEEE International Conference on Web Services (ICWS)*, 2018, pp. 187–194.

[11] J. Nam, J. Seo, and S. Shin, "Probius: Automated approach for vnf and service chain analysis in software-defined nfv," in *Proceedings of the Symposium on SDN Research*, 2018, pp. 1–13.

[12] R. Vilalta, A. Mayoral, R. Casellas, R. Martínez, and R. Muñoz, "Sdn/nfv orchestration of multi-technology and multi-domain networks in cloud/fog architectures for 5g services," in *2016 21st OptoElectronics and Communications Conference (OECC) held jointly with 2016 International Conference on Photonics in Switching (PS)*. IEEE, 2016, pp. 1–3.

[13] R. V. Rosa, M. A. S. Santos, and C. E. Rothenberg, "Md2-nfv: The case for multi-domain distributed network functions virtualization," in *2015 International Conference and Workshops on Networked Systems (NetSys)*. IEEE, 2015, pp. 1–5.

[14] K. D. Joshi and K. Kataoka, "psmart: A lightweight, privacy-aware service function chain orchestration in multi-domain nfv/sdn," *Computer Networks*, vol. 178, p. 107295, 2020.

[15] K. Katsalis, N. Nikaein, and A. Edmonds, "Multi-domain orchestration for nfv: Challenges and research directions," in *2016 15th International Conference on Ubiquitous Computing and Communications and 2016 International Symposium on Cyberspace and Security (IUCC-CSS)*. IEEE, 2016, pp. 189–195.

[16] J. C. Cisneros, S. Yangui, S. E. P. Hernandez, and K. Drira, "A survey on distributed nfv multi-domain orchestration from an algorithmic functional perspective," *IEEE Communications Magazine*, 2022.

[17] P. T. A. Quang, A. Bradai, K. D. Singh, and Y. Hadjadj-Aoul, "Multi-domain non-cooperative vnf-fg embedding: A deep reinforcement learning approach," in *IEEE INFOCOM 2019-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE, 2019, pp. 886–891.

[18] T. Subramanya and R. Riggio, "Centralized and federated learning for predictive vnf autoscaling in multi-domain 5g networks and beyond," *IEEE Transactions on Network and Service Management*, vol. 18, no. 1, pp. 63–78, 2021.

[19] C. Chen, L. Nagel, L. Cui, and F. P. Tso, "Distributed federated service chaining: A scalable and cost-aware approach for multi-domain networks," *Computer Networks*, vol. 212, p. 109044, 2022.

[20] D. Andreoletti, T. Velichkova, G. Verticale, M. Tornatore, and S. Giordano, "A privacy-preserving reinforcement learning algorithm for multi-domain virtual network embedding," *IEEE Transactions on Network and Service Management*, vol. 17, no. 4, pp. 2291–2304, 2020.

[21] O. Gupta and R. Raskar, "Distributed learning of deep neural network over multiple agents," *Journal of Network and Computer Applications*, vol. 116, pp. 1–8, 2018.

[22] P. Vepakomma, O. Gupta, T. Swedish, and R. Raskar, "Split learning for health: Distributed deep learning without sharing raw patient data," *arXiv preprint arXiv:1812.00564*, 2018.

[23] M. Sundararajan, A. Taly, and Q. Yan, "Axiomatic attribution for deep networks," in *International conference on machine learning*. PMLR, 2017, pp. 3319–3328.

[24] A. Shrikumar, P. Greenside, and A. Kundaje, "Learning important features through propagating activation differences," in *International conference on machine learning*. PMLR, 2017, pp. 3145–3153.

[25] P. Linardatos, V. Papastefanopoulos, and S. Kotsiantis, "Explainable ai: A review of machine learning interpretability methods," *Entropy*, vol. 23, no. 1, p. 18, 2020.

[26] "Itu-ai-ml-in-5g-challenge/itu-ml5g-ps-032-kddi-naist-lsm," https://github.com/ITU-AI-ML-in-5G-Challenge/ITU-ML5G-PS-032-KDDI-naist-lsm.

[27] J. Kawasaki, G. Mouri, and Y. Suzuki, "Comparative analysis of network fault classification using machine learning," in *NOMS 2020-2020 IEEE/IFIP Network Operations and Management Symposium*. IEEE, 2020, pp. 1–6.