An Artificial Intelligent Aided Unified Network for Secure Beyond 5G Long Term Evolution [GA: 101096456]

Deliverable 5.4

NANCY Explainable AI Toolbox

Programme: HORIZON-JU-SNS-2022-STREAM-A-01-06

Start Date: 01 January 2023

Duration: 36 Months

Co-funded by the European Union

NANCY project has received funding from the Smart Networks and Services Joint Undertaking (SNS JU) under the European Union's Horizon Europe research and innovation programme under Grant Agreement No 101096456.

Document Control Page

Deliverable Name	NANCY Explainable AI Toolbox
Deliverable Number	D5.4
Work Package	WP5 ‘Security, Privacy, and Trust Mechanisms’
Associated Task	T5.5 ‘Enabling Explainable AI Functionalities’
Dissemination Level	Public
Due Date	31 May 2025 (M29)
Completion Date	28 May 2025
Submission Date	29 May 2025
Deliverable Lead Partner	MINDS
Deliverable Author(s)	Stathis Mavridopoulos (MINDS), Dimitrios-Christos Asimopoulos (MINDS), Dimitrios Balompas (MINDS), Nikolaos Ntampakis (MINDS), Konstantinos Tsiamitros (MINDS), Christos Vasilakis (MINDS), Stylianos Trevlakis (INNO), Lamprini Mitsiou (INNO), Eirini Gkarnetidou (INNO), Vasileios Kouvakis (INNO), Theodoros Tsiftsis (INNO), Karypidis Paris-Alexandros (SID), Fountoukidis Eleftherios (SID), Karamitsiou Thomai (SID), Lytos Anastasios (SID), Andronikidis Georgios (SID), Kyranou Konstantinos (SID), Niotis Georgios (SID), Michoulis Georgios (SID), Tziolas Georgios (SID), Panagiotis Sarigiannidis (UOWM), Thomas Lagkas (UOWM), Athanasios Liatifis (UOWM), Dimitrios Pliatsios (UOWM), Sotirios Tegos (UOWM), Anna Triantafyllou (UOWM), Nikolaos Mitsiou (UOWM), Vasiliki Koutsioumpa (UOWM), Pigi Papanikolaou (UOWM)
Version	1.0

Document History

Version	Date	Change History	Author(s)	Organisation
0.1	14/02/2025	Table of contents	Stathis Mavridopoulos, Dimitrios-Christos Asimopoulos, Dimitrios Balompas, Nikolaos Ntampakis, Konstantinos Tsiamitros, Christos Vasilakis	MINDS
0.2	20/03/2025	Sections 1, 2 & 3	Stathis Mavridopoulos, Dimitrios-Christos Asimopoulos, Dimitrios Balompas, Nikolaos Ntampakis, Konstantinos	MINDS

			Tsiamitros, Christos Vasilakis	
0.3	31/03/2025	Sections 4 & 5	Karypidis Paris-Alexandros, Fountoukidis Eleftherios, Karamitsiou Thomai, Lytos Anastasios, Andronikidis Georgios, Kyranou Konstantinos, Niotis Georgios, Michoulis Georgios, Tziolas Georgios	SID
0.4	18/04/2025	Sections 2.2.2 & 3.2.3	Stylianos Trevlakis, Lamprini Mitsiou, Eirini Gkarnetidou, Vasileios Kouvakis, Theodoros Tsiftsis	INNO
0.5	28/04/2025	All sections	Stathis Mavridopoulos, Dimitrios-Christos Asimopoulos, Dimitrios Balompas, Nikolaos Ntampakis, Konstantinos Tsiamitros, Christos Vasilakis	MINDS
0.6	05/05/2025	Contributions & revisions to section 3	Panagiotis Sarigiannidis, Thomas Lagkas, Athanasios Liatifis, Dimitrios Pliatsios, Sotirios Tegos, Anna Triantafyllou, Nikolaos Mitsiou, Vasiliki Koutsioumpa, Pigi Papanikolaou	UOWM
0.7	12/5/2025	INNO final contributions	Eirini Gkarnetidou, Theodoros Tsiftsis	INNO
0.8	19/5/2025	Addressed reviewers comments	Stathis Mavridopoulos, Dimitrios-Christos Asimopoulos, Dimitrios Balompas, Nikolaos Ntampakis, Konstantinos	MINDS

			Tsiamitros, Christos Vasilakis	
1.0	28/05/2025	Final version after quality revisions	Dimitrios Pliatsios, Anna Triantafyllou	UOWM

Internal Review History

Name	Organisation	Date
Francisco Javier de Vicente Gutierrez	NEC	07 May 2025
Dimitrios Pliatsos, Sotirios Tegos	UOWM	09 May 2025

Quality Manager Revision

Name	Organisation	Date
Dimitrios Pliatsios, Anna Triantafyllou	UOWM	28 May 2025

Legal Notice

The information in this document is subject to change without notice.

The Members of the NANCY Consortium make no warranty of any kind about this document, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose.

The Members of the NANCY Consortium shall not be held liable for errors contained herein or direct, indirect, special, incidental, or consequential damages in connection with the furnishing, performance, or use of this material.

Co-funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or SNS JU. Neither the European Union nor the SNS JU can be held responsible for them.

Table of Contents

List of Figures

List of Tables

List of Annex Figures

List of Acronyms

Acronym	Explanation
AI	Artificial Intelligence
AINQM	AI-based Network Quality Management
API	Application Programming Interface
ASL	American Sign Language
B5G	Beyond Fifth Generation
DT	Digital Twin
FL-IDS	Federated Learning Intrusion Detection Systems
GDPR	General Data Protection Regulation
Grad-CAM	Gradient-weighted Class Activation Mapping
LIME	Local Interpretable Model-agnostic Explanations
LLM	Large Language Model
SemCom	Semantic Communication
SHAP	SHapley Additive exPlanations
XAI	eXplainable Artificial Intelligence

Executive summary

D5.4 "NANCY Explainable AI Toolbox" presents the overall architecture, functionalities and components of the explainable Artificial Intelligence (XAI) Toolbox. It provides an overview of the methodologies and tools that support transparency and interpretability for decision-making beyond 5G (B5G) network functions. The main components of the NANCY XAI Toolbox (https://github.com/Sidroco-Holdings-Ltd/NANCY_Explainable_AI_Toolbox) are (i) the anomaly detection XAI Component, (ii) the outage prediction XAI Component and (iii) the semantic communications XAI Component. Furthermore, the deliverable presents the implementation of the centralized XAI dashboard, which provides access to global and local model explanations through visual and interactive interfaces.

The integration of a Large Language Model (LLM)-Powered Analysis Component is introduced, as a Dashboard's sub-component, which facilitates the transformation of technical outputs into natural language for better usability by non-experts. Additionally, the deliverable provides a comprehensive description of the optimization strategies that support stability and scalability. Overall, this deliverable contributes towards a trustworthy and understandable AI, addressing the needs of end-users in network management.

1. Introduction

1.1. Purpose of this Deliverable

The main scope of D5.4 “NANCY Explainable AI Toolbox” is the demonstration of results from Task 5.5 “Enabling Explainable AI Functionalities”. Specifically, D5.4 aims to present the architecture, implementation and integration of the NANCY Explainable Artificial Intelligence (XAI) Toolbox. Our explainable AI Toolbox will enable the overall transparency and trustworthiness of AI-driven network management in NANCY.

In this direction, through the principles of explainable AI, the AI-enabled self-healing and self-recovery processes will become simpler and understandable to network operators. As AI models are progressively adopted for tasks such as anomaly detection, outage prediction, and semantic communication optimization, the need for more understandable explanations of models' decisions arises. That need is vital in terms of interpretability in modern network systems management, particularly in the context of 5G and beyond.

D5.4 “NANCY Explainable AI Toolbox” demonstrates several key objectives of the NANCY Explainable AI Toolbox. In D5.4, a modular, scalable and interoperable architecture for the XAI Toolbox, ensuring seamless integration with existing network management systems and other components, is defined. Moreover, the three specialized XAI components in NANCY are documented, namely: (1) Anomaly Detection, (2) Outage Prediction and (3) Semantic Communications. The NANCY XAI Dashboard, a web-based application that stands as a comprehensive and interactive platform for visualizing and analyzing Explainable AI (XAI) outputs, is presented. Additionally, a more Human-Centric Interpretability approach is achieved through the integration of a Large Language Model (LLM)-powered explanation layer that transforms complex numerical outputs from SHAP into natural language insights. That way, outputs are accessible to non-expert users such as network technicians and business stakeholders. The integration of the XAI Toolbox with Federated Learning Intrusion Detection Systems (FL-IDS) and AI-based Network Quality Management (AINQM) systems is outlined.

The deliverable supports that the network management through the NANCY project is transparent, auditable, and actionable and that it aligns with ethical AI principles and regulatory requirements such as EU’s GDPR.

1.2. Relation with other Work Packages and Deliverables

D5.4 “Explainable AI Toolbox” is included in WP5 “Security, Privacy, and Trust Mechanisms”, which focuses on developing advanced security, privacy and trust mechanisms for beyond-5G networks. Specifically, D5.4 is the output of Task 5.5 and contributes by enabling explainability in AI-based decision systems supporting network management.

D5.4 is complementary to other deliverables such as D5.2 “NANCY Security and Privacy Distributed Blockchain-based Mechanisms” and D5.3 “Self-healing and Self-recovery Mechanisms”. Additionally, Task 5.5 is aligned with WP2, which defines the project’s use cases and system requirements. The requirements expressed in WP2 ensure that the tools designed in D5.4 are practical and relevant to real-world operators’ needs and use cases.

D5.4 “Explainable AI Toolbox” also takes into account the AI infrastructure and orchestration mechanisms developed in WP3 and builds on the architecture design of WP3. The XAI toolbox aligns with WP3's goals by explaining the outputs of AI models used for anomaly detection, outage prediction, and semantic communication.

1.3. Structure of the Document

The rest of the document is structured as follows:

- **Section 2 – Background and State of the Art** discusses the relevant state-of-art focusing on the foundations of explainable AI and the well-known methods and technologies.
- **Section 3 – NANCY XAI Toolbox Architecture** documents the architecture of the NANCY AI Toolbox and the involved components.
- **Section 4 – XAI Dashboard Implementation** presents the dashboard implementation of the XAI toolkit and its main functionalities.
- **Section 5 – Integration and Deployment** discusses the integration and deployment processes.
- **Section 6 – Conclusion and Outlook** concludes the deliverable.

2. Background and State of the Art

Modern network infrastructures have become increasingly complex, necessitating advanced management techniques powered by artificial intelligence and machine learning. However, the "black box" nature of many AI algorithms presents challenges for network operators who need to understand, trust, and effectively act upon AI-generated insights and recommendations.

2.1. Explainable AI Foundations in Network Management

Traditional network management has relied on rule-based systems and manual intervention which, while transparent, struggle to scale with the increasing complexity of modern networks, particularly in 5G and beyond [1]. The advent of AI-driven network management solutions has addressed many scalability and efficiency challenges but has, on the other hand, introduced a critical opacity problem that XAI seeks to resolve.

Explainable AI in network management is built upon several key foundational principles that distinguish it from standard AI approaches. The first of these is transparency, which allows AI-driven decisions to be understandable for human operators [2]. Transparency is important in mission-critical network equipment where incorrect decisions can lead to severe service interruptions. Interpretability is the second principle, which provides clear explanations of AI predictions and guidance in language that can be understood and implemented by network operators. The third principle concerns accountability, with clear lines of responsibility being provided for AI-driven tool decision-making, an issue that is of specific importance in telecommunication environments [3].

Network management presents specific challenges to the utilization of XAI compared to other domains. These are the dynamic nature of flows of network traffic, the heterogeneity of network protocols and devices and the need for real-time decision-making [4]. Commercial and regulation demands have driven the creation of XAI in network management standards that mandate transparency in automated decision-making. For instance, the European Union General Data Protection Regulation (GDPR) has provisions for "the right to explanation" of automated decisions affecting humans, which includes network management decisions affecting service delivery [5]. Existing research has shown that XAI approaches would actually advance network Management outcomes. For example, studies have shown that network operators can diagnose incidents much faster when they are provided with explainable AI insights compared to traditional black-box AI suggestions. Additionally, the lack of trust between human operators and AI has been shown to be reduced when explainability functionality is added [6].

2.2. XAI Methods and Technologies in Network Management

The field of XAI for network administration covers a broad range of techniques and technologies, each having its individual strengths and applications. This subsection explores the XAI techniques that have been successfully implemented on network management issues, with a particular focus on classification-based techniques as a result of their application in network anomaly detection and traffic classification tasks.

2.2.1. XAI approaches for Classification Methods in Network Management

Classification is one of the key network management operations that includes significant activities such as traffic classification, anomaly detection and identification of attack types. [7] The explainability of these classification models is to depend on and act according to their predictions. Several of the most significant XAI approaches have been particularly effective for classification methods in network management cases.

The SHapley Additive exPlanations (SHAP) [8] method has been very well received in network management applications since it is well-founded theoretically in cooperative game theory. SHAP values provide a comprehensive assessment of feature importance that respects local accuracy, missingness, and consistency properties and are therefore extremely well suited to explaining classifications of network traffic and anomalies. SHAP has been used well to explain other models' decisions in network management (Figure 1).

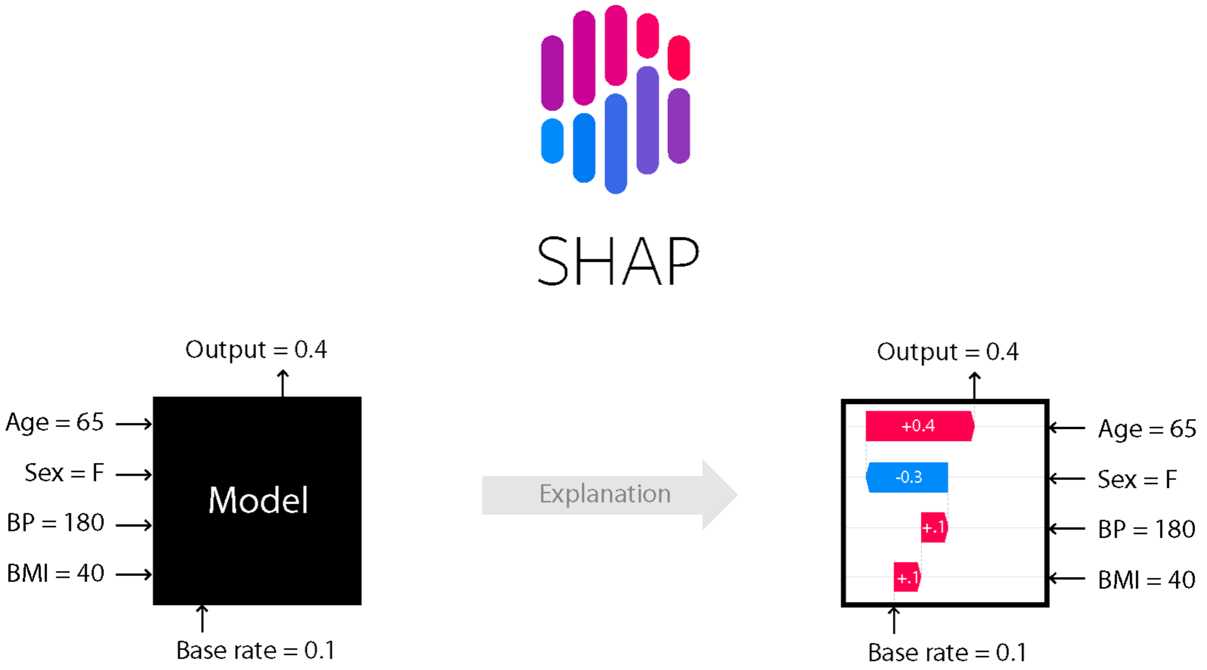


Figure 1: Indicative SHAP application

The application of SHAP to network traffic classification offers several advantages. First, it provides global insights into which features are most important across all classifications. This capability is particularly valuable in network security contexts, where operators need to understand both general attack patterns and specific incident details. Second, SHAP's ability to handle feature interactions makes it well-suited for network data, where correlations between features (e.g., packet size and protocol type) often contain significant predictive information. Third, visualization features of SHAP, such as force plots and summary plots are aligned with network operators' need for easy graphical representation of complex network behavior. Experimental tasks of SHAP on network management have proved SHAP to be effective for various tasks. For instance, Aljohani et. al.'s [9] experiment utilized SHAP to explain a deep learning model's decision process for DDoS attacks detection and discovered packet timing variance-related features were important in different advanced attacks.

LIME (Local Interpretable Model-agnostic Explanations [10]) is another prominent technique in XAI network management. LIME approximates a sophisticated model with a simpler, more interpretable model to explain individual predictions (Figure 2).

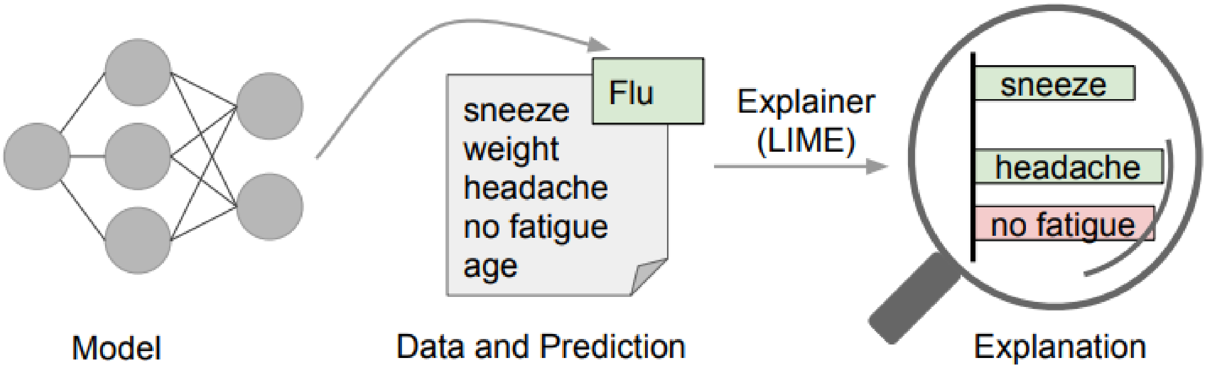


Figure 2: Indicative LIME application

This is useful in network management systems where explanations need to be provided to operators whose technical expertise is diverse. The interpretability of LIME's linear approximations allows its explanations to be understood by all, while its model-agnostic nature allows it to be applied directly to any classifier model being used in network management. LIME has already been used in numerous network management problems. For instance, it has been used to analyze intrusion detection system alarms to allow security experts to simply detect if alarms are real threats or not [11].

While both LIME and SHAP provide valuable explanations for network management classifications, they are two alternative compromises between the quality of explanation and the computational expense. LIME provides less computationally expensive results but potentially less informative explanations over comparable instances, while SHAP provides more theoretically robust explanations but at higher computation [12]. It is an important compromise for network management tasks where real-time explanatory capability might be really useful for operational decision-making.

Table 1: Comparison of XAI Methods for Network Traffic Classification

XAI Method	Explanation Type	Computational Complexity	Real-time Capability	Network Apps
SHAP	Global approximation	High	Limited	Anomaly explanation, Attack classification
LIME	Local approximation	Medium	Moderate	Intrusion detection, QoS classification

A comparison of XAI methods for network traffic classification is summarized in Table 1.The choice of the XAI method for classification in network management depends on several factors. Some of them are the type of classification problem, the model complexity, the working situation and the explanation consumer. Awareness of these trade-offs and selection of appropriate XAI methods is critical to creating practical explainable AI systems for network management.

2.2.2. XAI approaches for Semantic Communications

Semantic communication (SemCom) is also quickly emerging as a key field for building future systems, where systems seek to transmit not merely raw data, but the meaning intended behind said data. By focusing on the semantic rather than traditional bit-by-bit accuracy of information, SemCom can potentially very naturally complement the long-term vision for 5G and beyond (B5G) systems. Current studies point out the way SemCom offers the promise of novel methods for data encoding, sharing of context, and user-centric service adaptation, while reflecting the general trend of integration of intelligence-driven solutions in next-generation networks [13] [14]. This direction complements ongoing B5G developments by tailoring communication services to meet extreme performance requirements in applications like autonomous vehicles, healthcare, and immersive extended reality. Central to these advanced networks is the deployment of sophisticated artificial intelligence (AI) techniques. While SemCom is a very promising technology, the current research landscape reveals complex obstacles in realizing and, by extension, utilizing fully functional SemCom systems, particularly in areas of algorithmic transparency, model training complexity, and practical signal transmission [15]. In addition, the stringent requirements for parallel model building on both reception and transmission ends pose tremendous technical challenges, particularly with regard to varied application cases. The gap between theoretical SemCom models and existing infrastructure further aggravates the situation, making it increasingly challenging to solve these issues, thus making more interpretable and flexible methodology inevitable that can fit in comfortably within dominant technological ecosystems.

To address these multi-faceted challenges, the field of XAI provides a new path for overcoming the very limitations of current SemCom systems. Techniques under XAI provide the promise of a breakthrough by offering instruments for deciphering and interpreting the convoluted frequently opaque mechanisms that drive semantic content conveyance. By injecting transparency and interpretability into SemCom models, we aim to transform such systems from impenetrable black boxes to transparent and responsible communication patterns, open to systematized examination, improvement, and trust in different technological application domains. XAI for image-based SemCom attempts to address fundamental problems of transparency and interpretability undertaken visually motivated information processing. As deep learning models become more complex, it has become important to understand how these systems represent, extract, and convey semantic information. These XAI techniques attempt to disentangle the intricate decision-making process of neural networks and therefore allow insights into semantic generation, transformation and reconstruction representations.

In the context of image transmission, XAI techniques aim to bridge the gap between the black box neural network processing and semantic understanding on human terms. This includes the designing mechanisms which break down intricate image representations into understandable semantic units systematically, offer explicit explanations of the manner in which particular features are extracted and prioritized, enable consistent checking of semantic information transfer, and determine that the SemCom process is transparent and accountable. In [16] a trustworthy image SemCom framework that exemplifies XAI principles is introduced. Their approach creates an innovative image semantic encoder that transforms images into multiple explainable semantic representations, including natural language descriptions, semantic segmentation maps, and object-specific sub-images. By generating these representations in discrete data formats, the framework ensures compatibility with existing digital communication systems while providing unprecedented interpretability.

Building upon the foundational principles of explainability established in image SemCom, the domain of text-based SemCom has its own specific challenges and opportunities. While visual semantics can be reduced to segmentation maps and object representation, textual semantics require a more complex methodology of interpretation and transmission. The intrinsic complexity of human language, with its contextual dependencies, metaphorical expressions, and subtle semantic variations, requires sophisticated XAI techniques that can unravel the intricate layers of importance or meaning embedded within textual content [17]. In [18] a new SemCom system is introduced, which tries to disentangle textual representations into semantically interpretable components, thus presenting an approach to solving the systematic extraction and conveyance of semantic information. Their work highlighted the potential of creating more transparent text communication models that go beyond traditional black-box approaches. Subsequent research presented in [19] [20] proposed alternative methodologies using knowledge graphs as a means of representing text semantic information. These approaches sought to decompose textual information into a more structured form, and potentially more interpretable method of semantic representation.

In addition, semantic representations based on graphs have been an efficient method for encoding intricate relational data. The problem of representing difficult network topologies into understandable and transmissible semantic information has pushed researchers to develop more advanced XAI methods for graph-based communication networks. In that regard, [21] presents a novel scene graph-based SemCom model that overcomes the limitations of current global semantic extraction methods. With the use of scene graph representations, the approach offers a more explanatory and informative means of semantic transmission. The system encourages the recovery of scene graph semantics, which can be encoded in the form of informative graph embeddings, thus offering a more advanced means of encoding and transmission of complex relational information.

Part of this deliverable focuses on this research area by presenting XAI techniques for improving the interpretability, efficiency, and credibility of SemCom systems. For this, we address critical challenges in semantic information extraction, transmission, and reconstruction across two distinct scenarios. In the first scenario, we extract detailed explanations of the American sign language (ASL) SemCom framework that was presented in D4.4 [22]. The used techniques (SHAP and Grad-CAM) provide complementary explanations regarding how semantic information is extracted and weighted in the communication system. Grad-CAM provides visual explanations by coloring the most relevant areas in an image that contribute to a specific classification decision, and SHAP values provide a game-theoretic explanation technique for the impact of one feature on the model prediction. For the case of the second scenario, we examine a SemCom system integrated with a digital twin (DT) architecture for pedestrian object detection. For the second application, we utilized a YOLO model, for multi-body object recognition in real-time in several cameras to collect and transmit useful semantic information such as bounding boxes and object identification. Grad-CAM is used to provide transparency by identifying the significant image areas that affect detection choices and thus reduce bandwidth usage at the expense of not reducing communication efficiency. With these XAI methods, our work presents an invaluable step toward more interpretable, effective, and reliable SemCom systems. We emphasize that explainable AI can transform SemCom as a black-box process and make it a clear, transparent methodology that can be rigorously analyzed and optimized.

3. NANCY XAI Toolbox Architecture

The NANCY XAI Toolbox represents a comprehensive solution for bringing explainability to AI-driven network management. It tackles the essential requirement for transparency in ever more autonomous networking setups, especially in 5G and beyond scenarios. This chapter outlines the architecture of the NANCY XAI Toolbox, explaining its high-level design, component interaction and implementation specifics. The repository for the NANCY XAI Toolbox is located at: https://github.com/Sidroco-Holdings-Ltd/NANCY_Explainable_AI_Toolbox

3.1. High-Level Architecture

The architectural design of the NANCY Explainable AI (XAI) Toolbox (Figure 3) follows the modular design based on the three basic XAI components, and each of the components is modelled to explain aspects of network functions and security. The Anomaly Detection XAI Component is the first component that explains network traffic anomaly detection models and cyber-attack detection on various attack vectors. This component takes the network flow data as input and generates explanations for the identified threats, so as to enable security experts to identify the precise indicators that have triggered alerts. The second pillar is the Outage Prediction XAI Component, which enables transparency for models estimating future service outages in 5G networks based on performance measurements and resource allocation trends. The component supports proactive network management and maintenance of service level agreement. The third primary pillar is the Semantic Communications XAI Component, which addresses interpretability for semantic communications. All of these components work together through a centralized visualization dashboard that serves as the shared interface for human operators. Stakeholders use the dashboard to explore both global model behavior and individual prediction explanations. Augmenting the technical explanations generated by the XAI components, a Large Language Model (LLM) subsystem translates complicated feature importance values and technical jargon to natural language reasoning that is interpretable by users who do not have specialized machine learning or network engineering expertise. The design of the architecture can also integrate channels through which the XAI components can interact with other external specialized systems. Specifically, the Anomaly Detection XAI Component interacts with FL-IDS so that model explanation is enabled for models developed in privacy-preserving federated learning environments. Similarly, the Outage Prediction XAI Component interacts with AINQM to interpret predictions generated by quality management algorithms.

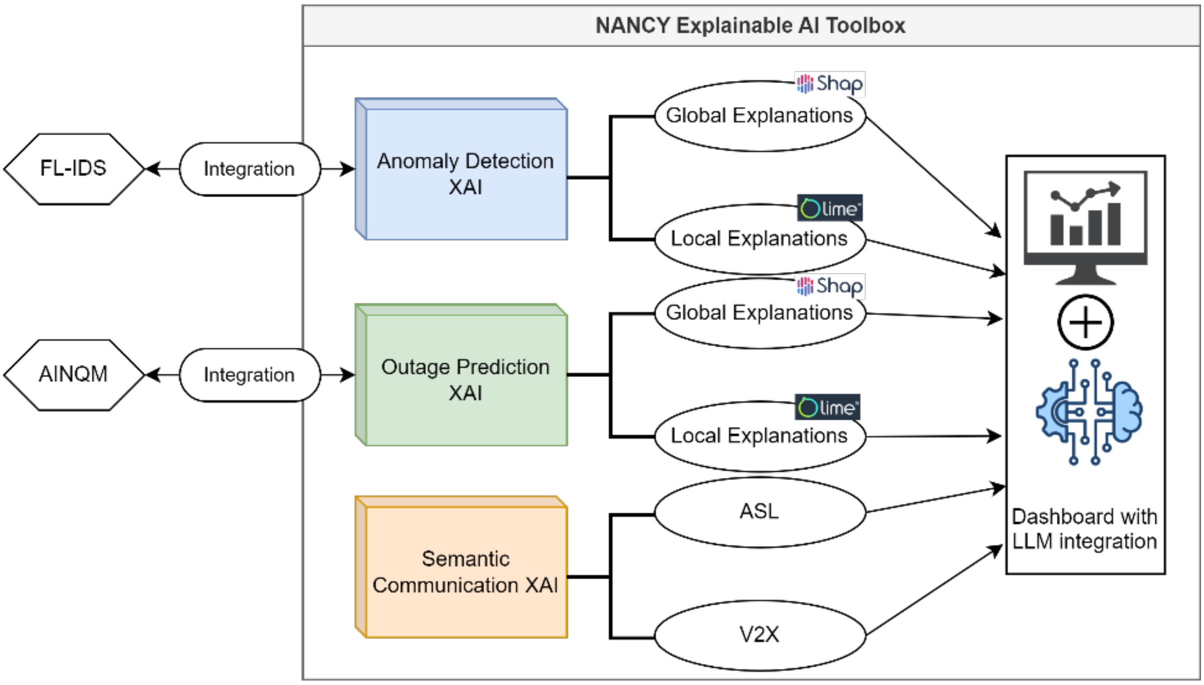


Figure 3: NANCY Explainable AI Toolbox High Level Architecture

The working pipeline starts with data from many network systems to be ingested by AI models. The models provide the predictions, which are then processed by the appropriate XAI component in order to create technical explanations. The explanations can continue to be visualized through the dashboard or upgraded via the LLM module for improved interpretability. Such constructed general explanations are then fed back to stakeholders in natural language descriptions and visualizations.

3.2. Component Overview and Interactions

The NANCY XAI Toolbox components interact within an ecosystem that enables explainability across different network domains. These components implement approaches to explanation generation, employing techniques appropriate to their specific domains while maintaining a consistent output format that facilitates integration with the visualization dashboard and LLM-based interpretation layer. The following sections provide detailed technical specifications of each component, their internal mechanisms, and their contextual position within the broader NANCY XAI framework.

3.2.1. Anomaly Detection XAI Component

The Anomaly Detection XAI Component represents an implementation of explainable artificial intelligence designed to elucidate the decision-making processes of intrusion detection models. This component specializes in interpreting the predictions of classification models trained to categorize network traffic flows into seven distinct traffic types: Benign Traffic, Reconnaissance Attack, TCP Scan, SYN Scan, SYN Flood, HTTP Flood, and Slowrate DoS. Data used for the implementation of anomaly detection XAI component was gathered under Greek in-door testbed according to D6.2. The technical implementation harnesses explanation techniques to provide interpretability at both global and local levels. For global model interpretability, the component implements SHAP methodology through the *TreeExplainer class*. The global explanation process commences with intelligent data sampling, selecting up to *5000* representative instances to balance computational efficiency with explanation accuracy. The sample data undergoes transformation into SHAP values, which quantify the contribution of each feature to predictions across all classes. These values are rendered into visualization artifacts and structured data formats to be consumed later by Dashboard and LLM components.

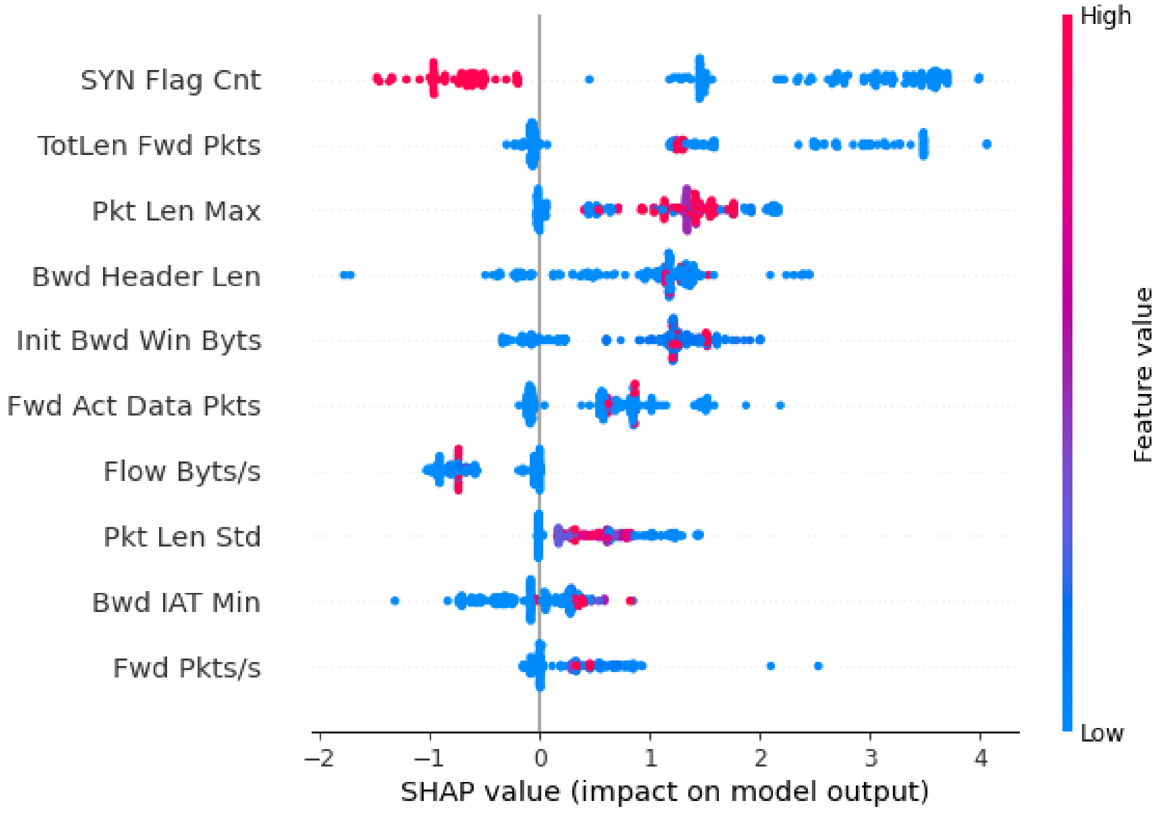


Figure 4: Indicative Example No.1 for Global Explanation under Anomaly Detection [Benign Traffic]

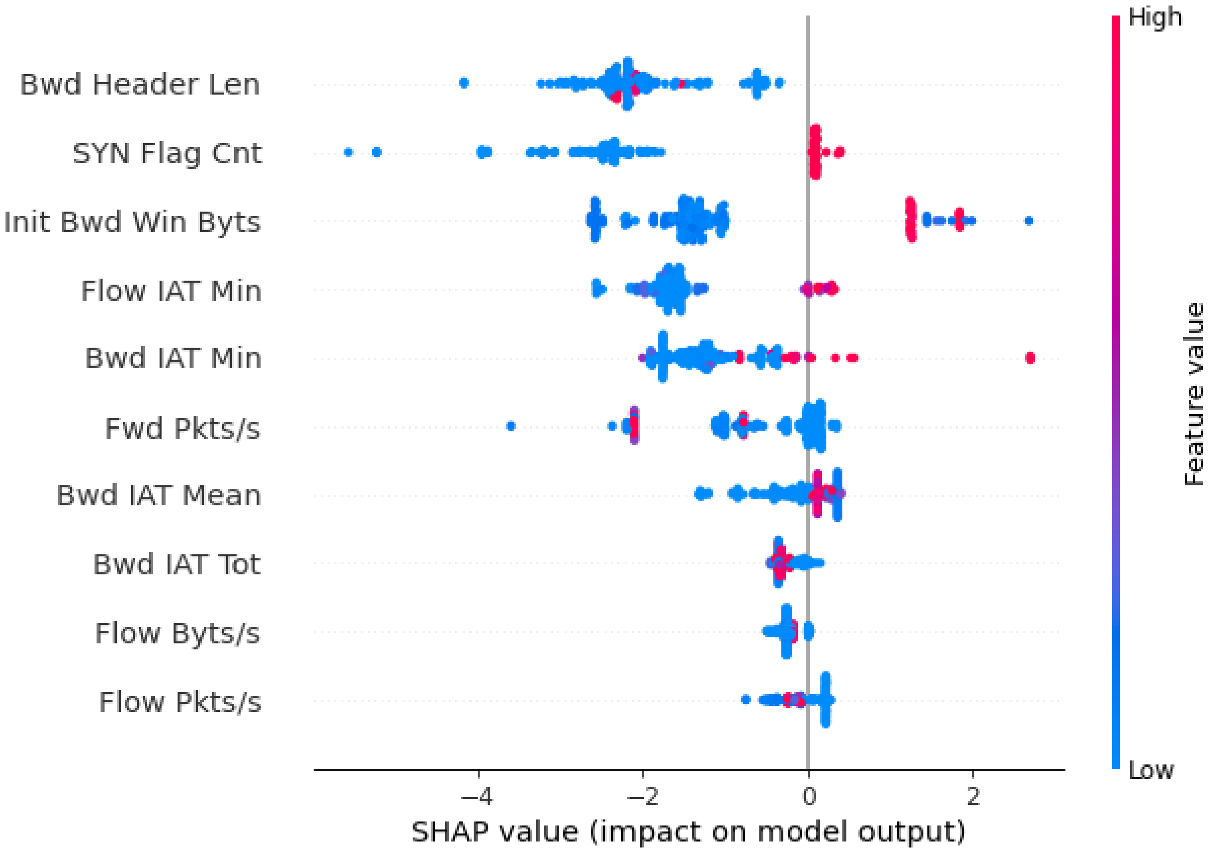
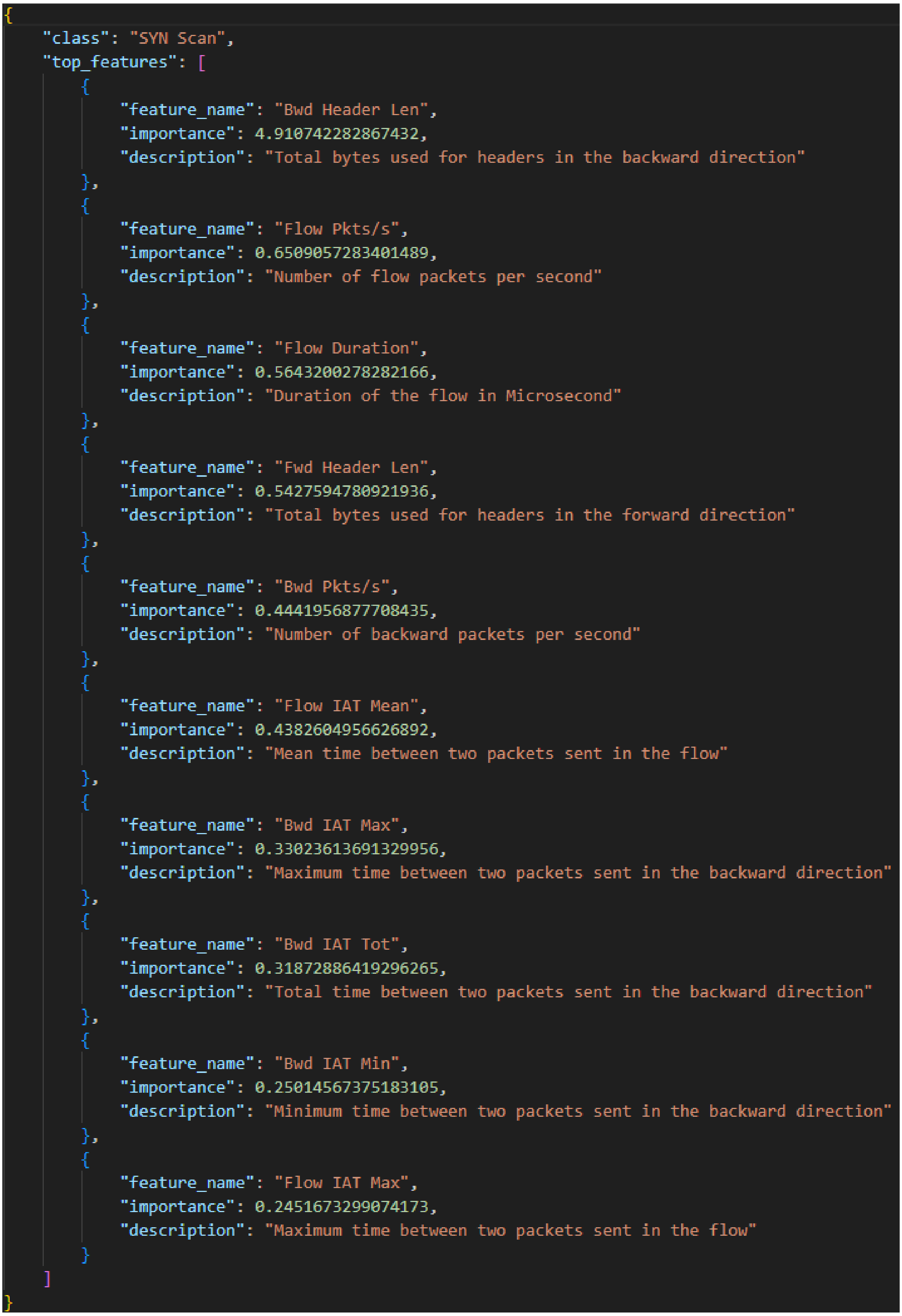


Figure 5: Indicative Example No.2 for Global Explanation under Anomaly Detection [SYN_Flood]

The visualizations manifest as summary plots with dot representations indicating both magnitude and direction of feature impacts. Figure 4 and Figure 5 display SHAP values showing how network traffic features influence the model. Figure 4 represents benign traffic, where features like SYN Flag Count, TotLen Fwd Pkts, and Pkt Len Max have varied impacts on the model output, with values distributed across both positive and negative ranges. Figure 5 shows SYN_Flood attack traffic, where SYN Flag Count exhibits a more distinctive pattern with high values concentrated around positive SHAP values. Features like Init Bwd Win Byts and Bwd IAT Min also show a clearer separation between high and low values. This difference makes sense since SYN_Flood attacks involve overwhelming targets with TCP SYN packets. Simultaneously, the component generates JSON structures containing detailed feature importance rankings, accompanied by natural language descriptions of each feature's semantic meaning within the network security context for dashboard integration.



```
{
    "class": "SYN Scan",
    "top_features": [
        {
            "feature_name": "Bwd Header Len",
            "importance": 4.910742282867432,
            "description": "Total bytes used for headers in the backward direction"
        },
        {
            "feature_name": "Flow Pkts/s",
            "importance": 0.6509057283401489,
            "description": "Number of flow packets per second"
        },
        {
            "feature_name": "Flow Duration",
            "importance": 0.5643200278282166,
            "description": "Duration of the flow in Microsecond"
        },
        {
            "feature_name": "Fwd Header Len",
            "importance": 0.5427594780921936,
            "description": "Total bytes used for headers in the forward direction"
        },
        {
            "feature_name": "Bwd Pkts/s",
            "importance": 0.4441956877708435,
            "description": "Number of backward packets per second"
        },
        {
            "feature_name": "Flow IAT Mean",
            "importance": 0.4382604956626892,
            "description": "Mean time between two packets sent in the flow"
        },
        {
            "feature_name": "Bwd IAT Max",
            "importance": 0.33023613691329956,
            "description": "Maximum time between two packets sent in the backward direction"
        },
        {
            "feature_name": "Bwd IAT Tot",
            "importance": 0.31872886419296265,
            "description": "Total time between two packets sent in the backward direction"
        },
        {
            "feature_name": "Bwd IAT Min",
            "importance": 0.25014567375183105,
            "description": "Minimum time between two packets sent in the backward direction"
        },
        {
            "feature_name": "Flow IAT Max",
            "importance": 0.2451673299074173,
            "description": "Maximum time between two packets sent in the flow"
        }
    ]
}
```

Figure 6: Indicative JSON output for Global Explanation under Anomaly Detection [SYN_Scan]

Local interpretability is achieved through the implementation of the LIME technique. The LIME algorithm creates locally faithful surrogate models that approximate the complex decision boundary of the original model in the vicinity of specific instances. This process involves generating perturbed samples around the target instance, obtaining predictions for these samples, and fitting a simple interpretable model such as a linear regressor to the perturbed dataset. The coefficients of this surrogate model reveal feature contributions to the specific prediction. The component visualizes these contributions as horizontally oriented bar charts indicating both the magnitude and direction of feature influence. The local explanation artifacts are stored both as visualizations and as structured JSON data containing feature contributions and descriptions for dashboard integration. For instance, Figure 7 for Flow_ID_1 with a prediction of TCP_Scan revealed that the most important feature is Bwd PSH Flags. Similarly, for Flow_ID_373 in Figure 8 revealed that the most important feature for the prediction of Benign traffic in this specific example was Bwp_Pkts/s. Figure 9 illustrates the JSON output from the component.

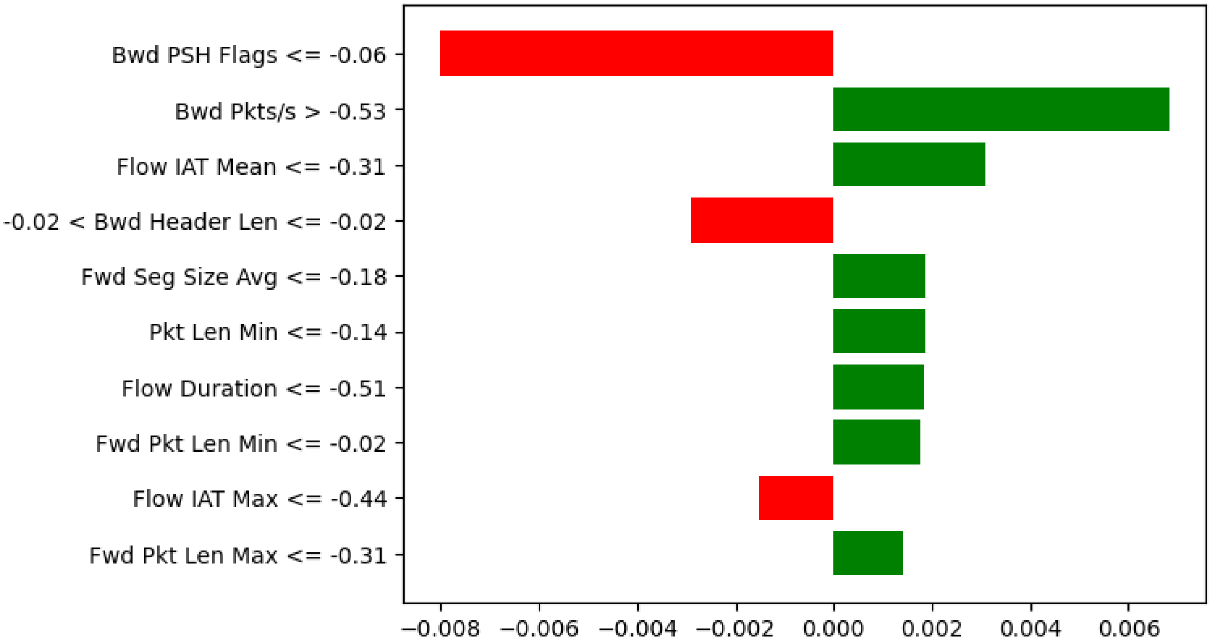


Figure 7: Indicative Example No.1 for Local Explanation under Anomaly Detection [Flow_ID_1_Actual_TCP_Scan_Predicted_TCP_Scan]

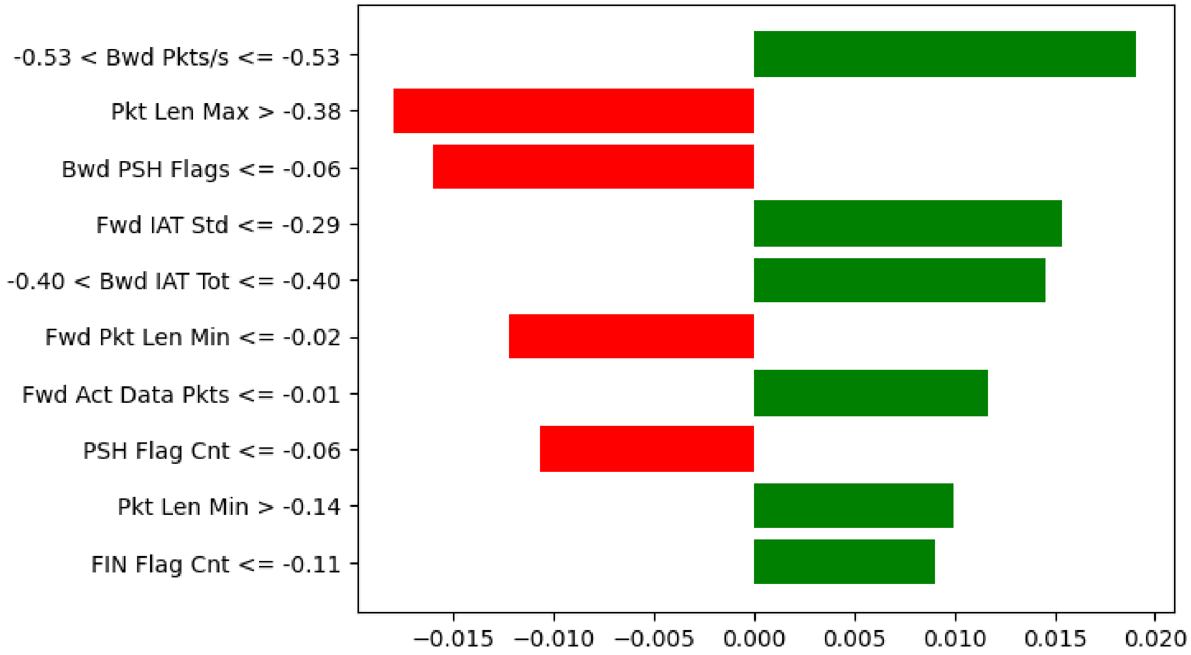
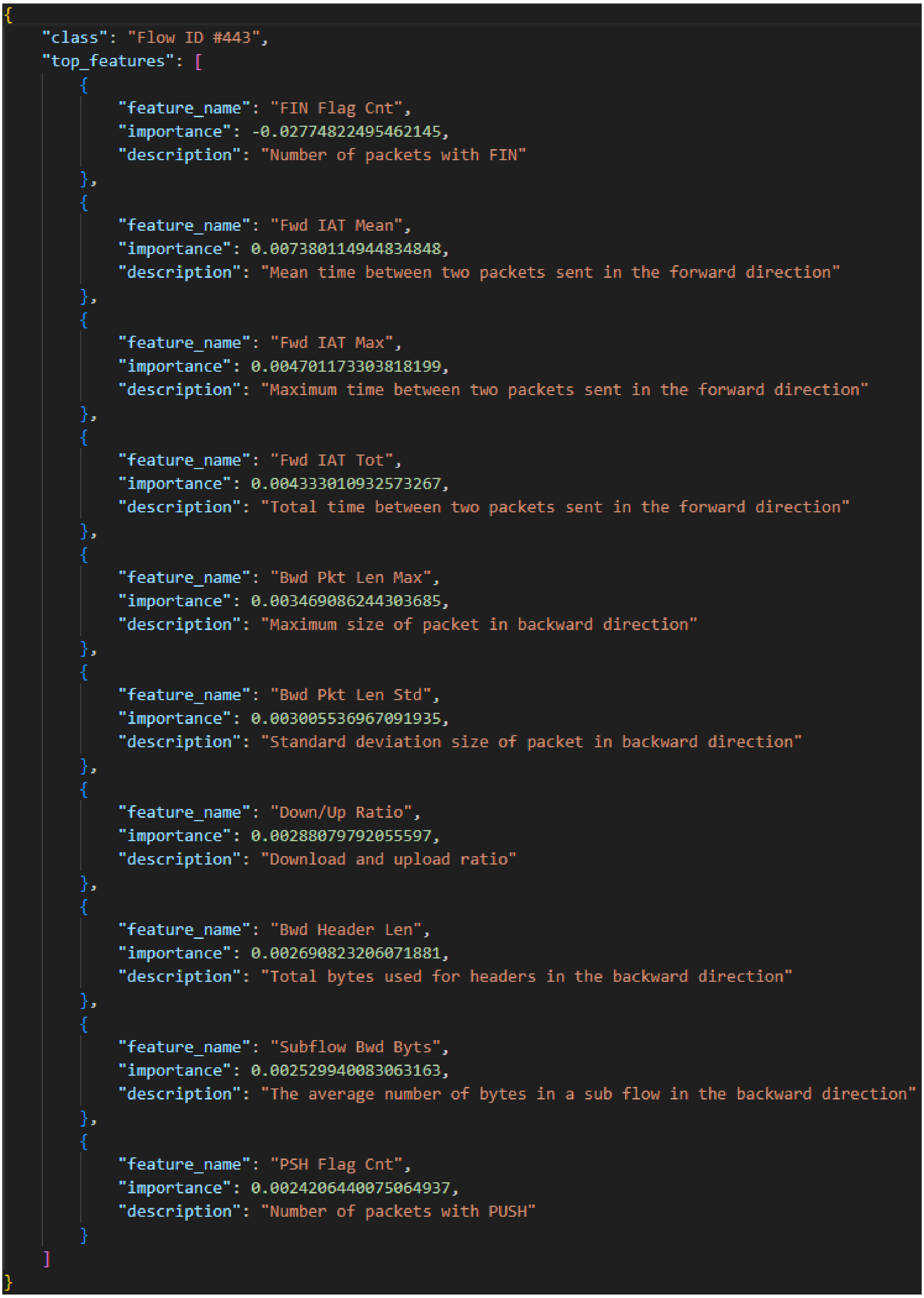


Figure 8: Indicative Example No.2 for Local Explanation under Anomaly Detection [Flow_ID_373_Actual_Benign_Traffic_Predicted_Benign_Traffic]



```
{
    "class": "Flow ID #443",
    "top_features": [
        {
            "feature_name": "FIN Flag Cnt",
            "importance": -0.02774822495462145,
            "description": "Number of packets with FIN"
        },
        {
            "feature_name": "Fwd IAT Mean",
            "importance": 0.007380114944834848,
            "description": "Mean time between two packets sent in the forward direction"
        },
        {
            "feature_name": "Fwd IAT Max",
            "importance": 0.004701173303818199,
            "description": "Maximum time between two packets sent in the forward direction"
        },
        {
            "feature_name": "Fwd IAT Tot",
            "importance": 0.004333010932573267,
            "description": "Total time between two packets sent in the forward direction"
        },
        {
            "feature_name": "Bwd Pkt Len Max",
            "importance": 0.003469086244303685,
            "description": "Maximum size of packet in backward direction"
        },
        {
            "feature_name": "Bwd Pkt Len Std",
            "importance": 0.003005536967091935,
            "description": "Standard deviation size of packet in backward direction"
        },
        {
            "feature_name": "Down/Up Ratio",
            "importance": 0.00288079792055597,
            "description": "Download and upload ratio"
        },
        {
            "feature_name": "Bwd Header Len",
            "importance": 0.002690823206071881,
            "description": "Total bytes used for headers in the backward direction"
        },
        {
            "feature_name": "Subflow Bwd Byts",
            "importance": 0.002529940083063163,
            "description": "The average number of bytes in a sub flow in the backward direction"
        },
        {
            "feature_name": "PSH Flag Cnt",
            "importance": 0.0024206440075064937,
            "description": "Number of packets with PUSH"
        }
    ]
}
```

Figure 9: Indicative JSON output for Local Explanation under Anomaly Detection [Flow_ID_443_Actual_HTTP_Flood_Predicted_HTTP_Flood]

The data preprocessing pipeline within the component implements handling of network traffic data. The process begins with elimination of non-feature columns including *Flow ID, IP addresses, port numbers, protocol identifiers, timestamps, and labels*. The resulting feature set undergoes cleaning operations to address missing values and infinite entries through appropriate replacement and filtering. Subsequently, the data is standardized using a pre-trained scaler that normalizes feature distributions to ensure algorithmic stability and consistent interpretation. This preprocessing ensures that explanation algorithms receive clean, normalized data properly formatted for explanation generation.

Table 2 provides a summary of the parameters employed by the Anomaly Detection XAI Component:

Table 2: Anomaly Detection XAI Component Parameters

Parameter	Value	Description
Scaler Type	*StandardScaler*	Centers features at mean 0 with unit variance
Missing Value Strategy	Removal	Rows with NaN values are excluded
SHAP Sampling Size	5000	Maximum samples used for global explanation generation
LIME Perturb Strategy	Discretization	Continuous features are discretized for perturbation
Local Explanation Features	10	Number of top features shown in local explanations

The component maintains integration capability with FL-IDS, allowing explanation of models trained through privacy-preserving federated learning approaches. This integration occurs through a model transfer mechanism that ensures compatibility between federated model outputs and the XAI component's explanation algorithms. The integrity is ensured through various functional and integration tests.

3.2.2. Outage Prediction XAI Component

The Outage Prediction XAI Component constitutes an implementation of explainable artificial intelligence techniques targeted at interpreting predictions related to service disruptions in 5G network environments. This component focuses on binary classification decisions that categorize network states into Normal Operation (transmission rates ≥ 0.01 Mbps) or Outage Risk (transmission rates < 0.01 Mbps) based on performance metrics. The component operates on a specific feature space identical to AINQM component: downlink buffer size in bytes (*dl_buffer*), number of transmitted packets in the downlink direction (tx_pkts), Channel Quality Indicator for downlink (dl_cqi), sum of requested Physical Resource Blocks (*sum_requested_prbs*), and sum of granted Physical Resource Blocks (*sum_granted_prbs*).

For global model understanding, the component implements the SHAP methodology through *TreeExplainer*, which calculates attribution values by analyzing all possible feature coalitions. The SHAP implementation within this component handles the binary classification nature of outage prediction by generating separate explanation sets for Normal Operation and Outage Risk classes. The visualization approach employs distinctive color gradients to emphasize feature contributions, with blue tones indicating factors contributing to normal operation and red tones highlighting outage risk factors. The global explanations provide network operations personnel with comprehensive understanding of model behavior across different network conditions and enable identification of systemic vulnerabilities. Also, the relevant JSON file is produced, used for dashboard integration. Figure 10 and Figure 11 show the SHAP values for ‘Normal Operation’ and ‘Outage Risk’, while Figure 12 shows the global explanation under outage prediction.

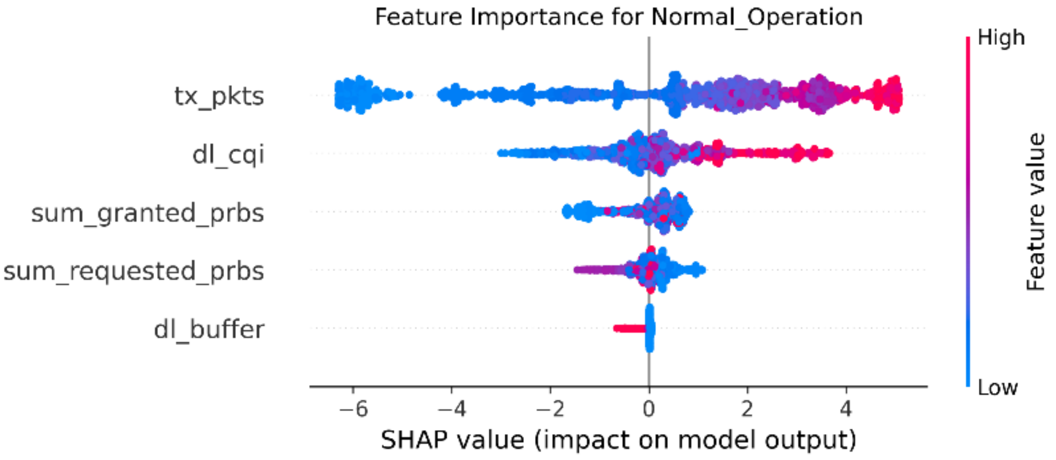


Figure 10: Global Explanation under Outage Prediction [Normal_Operation]

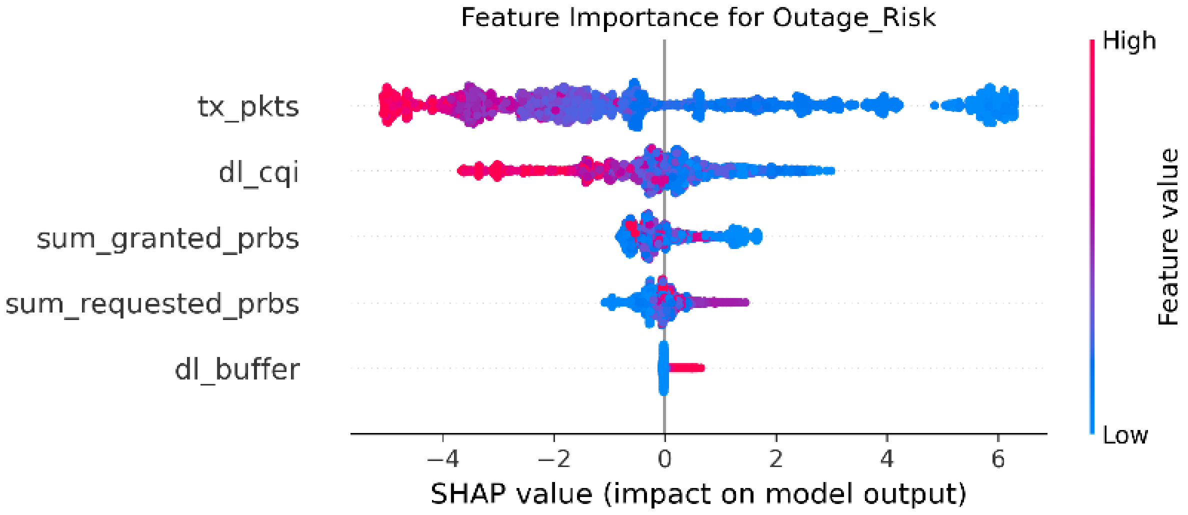
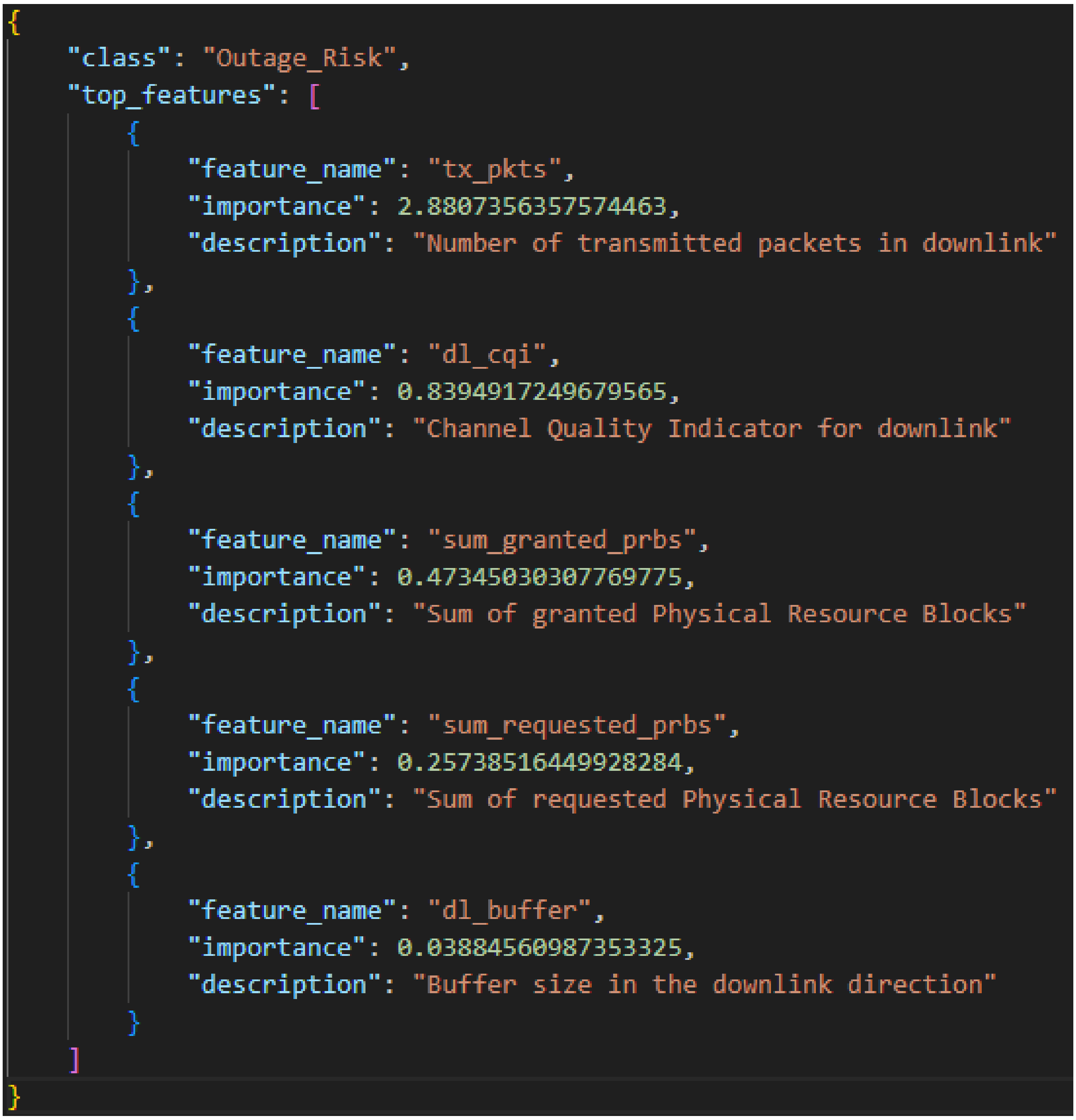


Figure 11: Global Explanation under Outage Prediction [Outage_Risk]



```
{
    "class": "Outage_Risk",
    "top_features": [
        {
            "feature_name": "tx_pkts",
            "importance": 2.8807356357574463,
            "description": "Number of transmitted packets in downlink"
        },
        {
            "feature_name": "dl_cqi",
            "importance": 0.8394917249679565,
            "description": "Channel Quality Indicator for downlink"
        },
        {
            "feature_name": "sum_granted_prbs",
            "importance": 0.47345030307769775,
            "description": "Sum of granted Physical Resource Blocks"
        },
        {
            "feature_name": "sum_requested_prbs",
            "importance": 0.25738516449928284,
            "description": "Sum of requested Physical Resource Blocks"
        },
        {
            "feature_name": "dl_buffer",
            "importance": 0.038845609873533325,
            "description": "Buffer size in the downlink direction"
        }
    ]
}
```

Figure 12: Indicative JSON output for Global Explanation under Outage Prediction [Outage_Risk]

Local explanations are generated through a LIME implementation which incorporates domain-specific adaptations for outage prediction. The component implements a custom prediction probability function, ensuring that explanations accurately reflect the operational decision boundary. This threshold adjustment addresses the inherent imbalance in outage scenarios, where false negatives (missed outage predictions) carry significantly higher operational cost than false positives. The local explanation process generates sample-specific visualizations that highlight the particular network conditions contributing to an outage prediction, enabling targeted remediation actions, along with the JSON outputs for dashboard integration. Figure 13 and Figure 14 show two examples for local explanation, while the respective JSON output is shown in Figure 15.

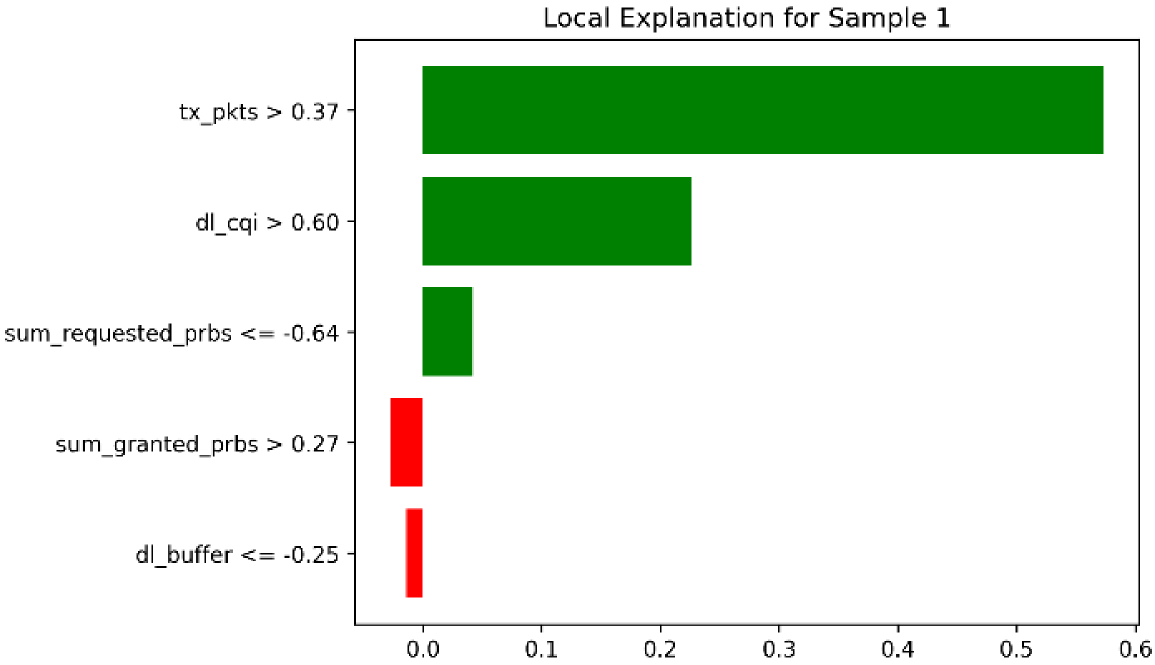


Figure 13: Indicative Example No.1 for Local Explanation under Outage Prediction [Sample_1_Actual_Normal_Operation_Predicted_Normal_Operation]

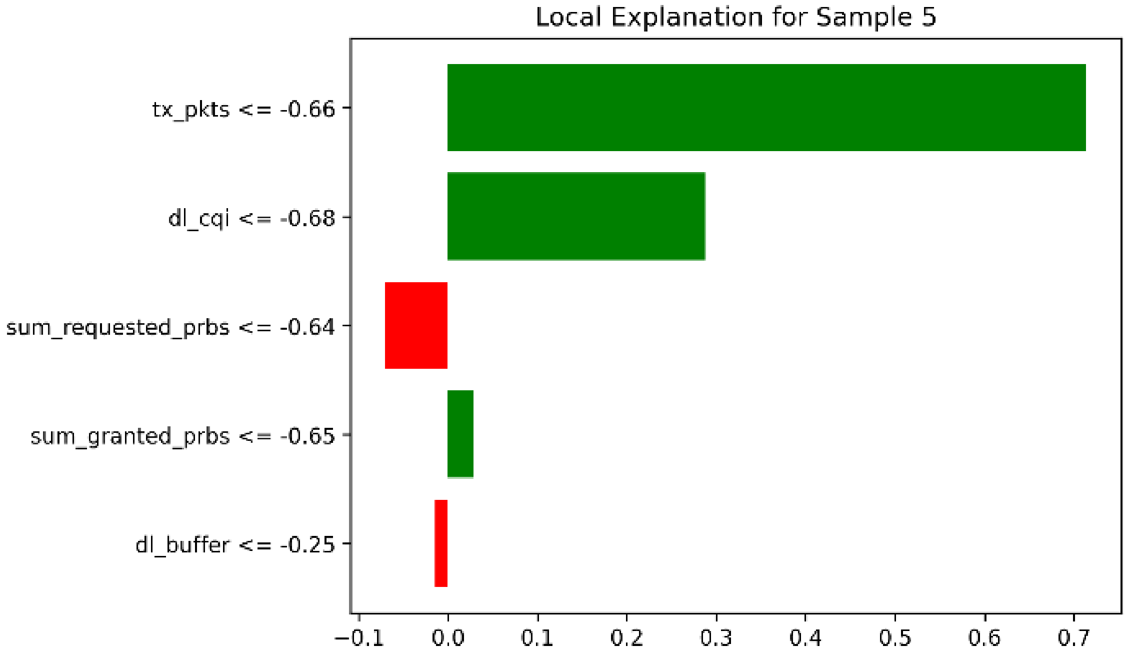
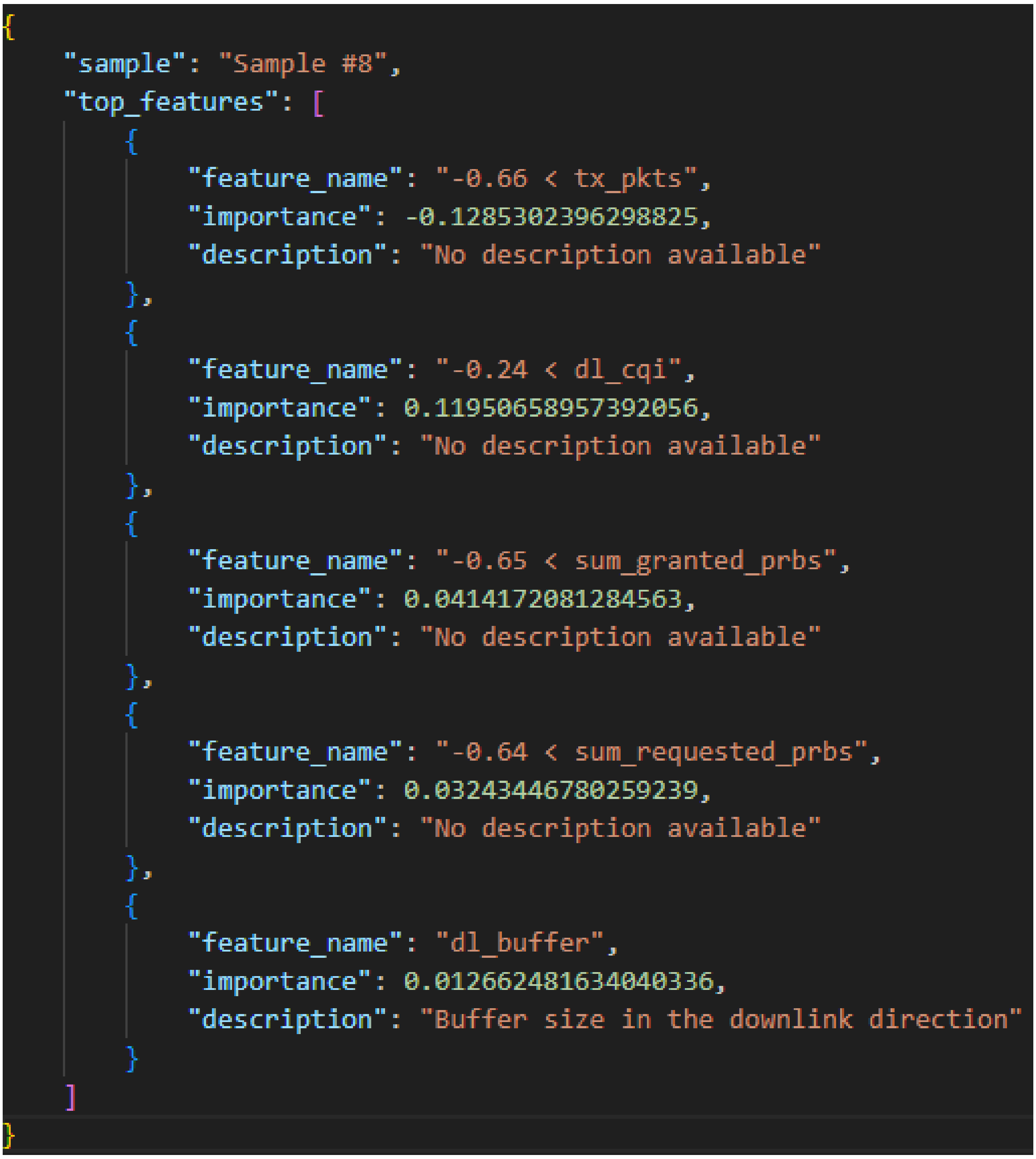


Figure 14: Indicative Example No.2 for Local Explanation under Outage Prediction [Sample_5_Actual_Outage_Risk_Predicted_Outage_Risk]



```
{
    "sample": "Sample #8",
    "top_features": [
        {
            "feature_name": "-0.66 < tx_pkts",
            "importance": -0.1285302396298825,
            "description": "No description available"
        },
        {
            "feature_name": "-0.24 < dl_cqi",
            "importance": 0.11950658957392056,
            "description": "No description available"
        },
        {
            "feature_name": "-0.65 < sum_granted_prbs",
            "importance": 0.0414172081284563,
            "description": "No description available"
        },
        {
            "feature_name": "-0.64 < sum_requested_prbs",
            "importance": 0.03243446780259239,
            "description": "No description available"
        },
        {
            "feature_name": "dl_buffer",
            "importance": 0.012662481634040336,
            "description": "Buffer size in the downlink direction"
        }
    ]
}
```

Figure 15: Indicative JSON output for Local Explanation under Outage Prediction [Sample_8_Actual_Normal_Operation_Predicted_Normal_Operation]

A special technical feature of this component is the column name standardization mechanism implemented through the *clean_column_names()* function. This function maps diverse feature naming conventions from different data sources to a standardized internal representation, ensuring explanation consistency across varying input formats. The standardization process handles variations in delimiter characters, unit notations, and directional indicators, converting them to a canonical form that facilitates both explanation generation and human interpretation. As the AINQM, the component was trained on the GitHub Colosseum Oran Dataset [23], specifically utilizing subsets containing comprehensive 5G network performance metrics including buffer utilization patterns, packet transmission statistics, channel quality indicators, and physical resource block allocation metrics.

Table 3 sums up the technical parameters and thresholds employed in the Outage Prediction XAI Component:

Table 3: Outage Prediction XAI Component Parameters

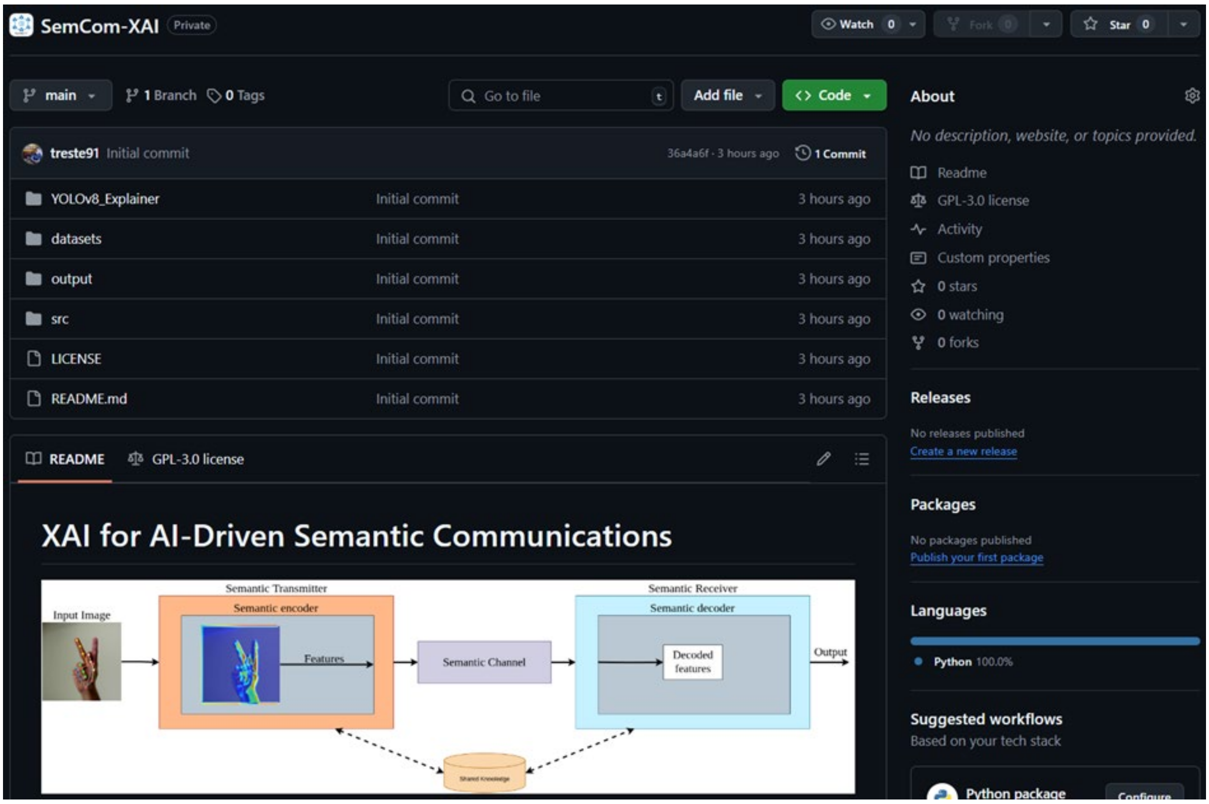
Parameter	Value	Description
Outage Threshold	< 0.01 Mbps	Transmission rate below which outage is defined (same as AINQM)
Classification Confidence	0.2 (20%)	Probability threshold for binary classification (same as AINQM)
SHAP Plot Type	Dot	Visualization format for global importance
LIME Discretization	Enabled	Continuous features are discretized for perturbation
Column Standardization	Mapping Function	Converts varied column names to standard format

The Outage Prediction XAI Component maintains integration capability with AINQM through a client-server architecture. This integration enables explanation of outage predictions generated by the quality management algorithms. Both the functionality of the XAI component and the integration with AINQM were checked through a series of functional and integration tests.

3.2.3. Semantic Communications XAI Component

Recent work on beyond 5G and emerging 6G networks has highlighted the importance of semantically enriched communication [24] [25]. By extracting and filtering goal-specific semantic information at the source and performing semantic decoding and post-processing at the destination, these systems aim to handle multiple time-varying, deadline-constrained traffic flows in a multi-user, distributed edge-to-cloud network. Consequently, new frameworks for semantic information extraction, novel knowledge representation models, and innovative metrics infused with semantics are required to manage congestion and measure performance while preserving relevancy.

Based on the above, we have created the SemCom-XAI repository on the official NANCY GitHub page, which is presented in Figure 16. This repository holds the implementation of XAI SemCom across two distinct scenarios, utilizing CNNs and the YOLO object detection framework.



treste91 Initial commit 36a4a6f · 3 hours ago 1 Commit

YOLOv8_Explainer	Initial commit	3 hours ago
datasets	Initial commit	3 hours ago
output	Initial commit	3 hours ago
src	Initial commit	3 hours ago
LICENSE	Initial commit	3 hours ago
README.md	Initial commit	3 hours ago

No description, website, or topics provided.

Readme
GPL-3.0 license
Activity
Custom properties
0 stars
0 watching
0 forks

Releases

No releases published
Create a new release

Packages

No packages published
Publish your first package

Languages

Python 100.0%

XAI for AI-Driven Semantic Communications

Figure 16: SemCom-XAI repository.

This project contains two primary scenarios:

1. ASL gesture recognition: Images of ASL gestures are processed by a CNN to extract only the semantic meaning of the sign. Instead of sending the entire image, the system transmits a condensed representation (semantic features) which can be interpreted on the receiving end. A dataset has been created within the NANCY project for ASL-based SemCom [26].

2. V2x (Vehicle-to-Everything) object detection: In a connected vehicle environment, real-time camera feeds are used with YOLO to detect and classify objects (vehicles, pedestrians, traffic signs, etc.). Again, the focus is on sending semantic information (e.g., object types and positions) rather than raw video frames, optimizing communication channels and reducing bandwidth. A dataset has been created within the NANCY project for V2x multi-view SemCom [27].

ASL Gesture recognition

In the first scenario, a system categorizes ASL signs within a SemCom model where, instead of sending the full raw image, only its meaning is sent. At a higher level of description, the process begins by obtaining raw gesture images, which are then preprocessed through resizing and normalization. These refined images are fed into a custom CNN that extracts important semantic features, transforming the visual data into a compact feature vector. This approach enhances efficiency by transmitting only key information through the network. This system not only identifies each gesture with high accuracy but also explains the reasoning behind each prediction through advanced XAI techniques.

One of the two chosen interpretability methods is GradCAM. This technique plays an important role in the visualization of the inner workings of the neural network. It has the ability to compute the gradients of the predicted class relative to the desired convolutional layer's feature maps and provides a heatmap that highlights the regions of the input image that contributed the most towards the prediction. As is evident from Figure 17, which shows three different gesture images, as well as four

GradCAM outputs from the last convolutional layer of the model for every image. For each layer, the GradCAM output is a single image where the heatmap is superimposed over the original hand sign and easily displaying the high attention areas of the raw image with warmer colors while it uses cooler colors for less important areas. The GradCAM visualization offers a macro-level representation by showing the overall areas of interest in the input image. It ensures that the network's focus is properly directed at the respective regions of the hand, making it easier to identify any errors in case the model ever glances at the wrong features. Visualization is important in developing confidence in the model because it enables the user to get an instantaneous, intuitive sense of what the network is "looking for" when making its predictions.

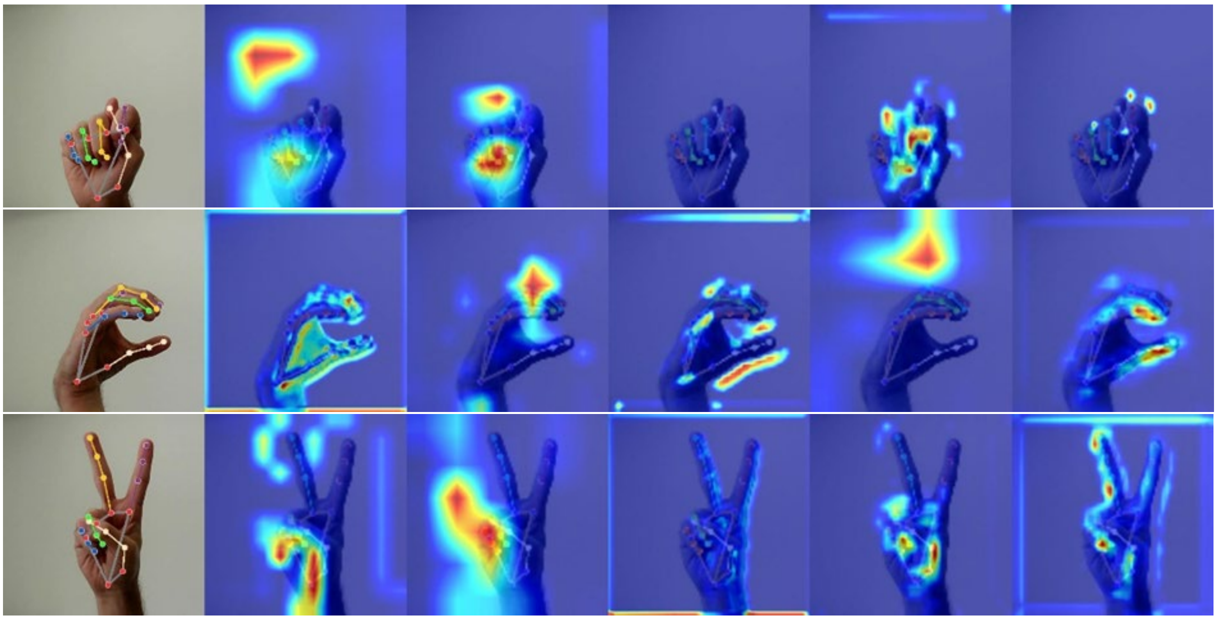


Figure 17: GradCAM heatmaps for different ASL letters.

In addition, in order to further increase the interpretability of the CNN model, the SHAP technique was applied to provide a more detailed explanation of the model predictions. This method uses Shapley values from game theory to calculate the contribution of each input attribute to the final decision. The visualization of SHAP is presented as a series of images: the first image shows the initial input (raw image), while the rest of the images (to the right) show the SHAP values for different regions of the image, as shown in Figure 18. Each of these images uses a color combination to indicate positive or negative contributions, clearly showing which features of the image drive the model towards a particular output category, and which ones drive the former away from it. The above technique provides a more detailed analysis, capable of showing even the effect of the smallest individual features of each image. By observing the original input image and the corresponding SHAP images, we can see how each segment or pixel group of the gesture contributes to the final class. This analysis not only helps to identify the most critical features but also makes the debugging process more accurate by highlighting any inconsistencies or incorrect feature mappings that may affect the performance of the model.

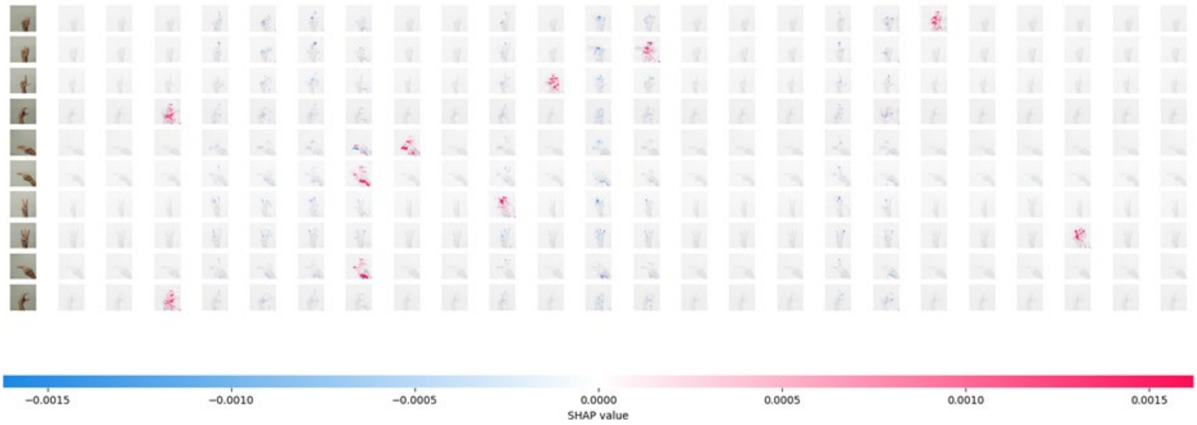


Figure 18: SHAP heatmaps for different ASL letters.

Together with one another, GradCAM and SHAP visualizations create a balanced XAI framework in the SemCom model. While GradCAM presents a broad, attention-based summary that is visually informative and description-ready in an instant, SHAP delves deeper into feature importance, presenting an elaborate explanation that facilitates analysis to a fine-grained extent and model optimization. This bidirectional approach not only enhances the system's transparency and trustworthiness but also makes it possible to continuously refine it through the revealing of the internal thinking of the model and rendering it comprehensible.

V2x object detection

In the second scenario, another SemCom model system is present, combined with a DT as the final product. First, we have a YOLO model that detects and classifies pedestrians in real time, analyzing streams from three cameras placed at different locations and angles but "looking" at the same point. Instead of sending full images, this model sends only the necessary semantic information, such as bounding boxes, object categories and other metadata, thus drastically reducing bandwidth usage and promoting fast and efficient communication.

The process starts by capturing raw images from all camera units. These images are fed into the YOLO detection model, which detects pedestrians in each frame. After extracting the necessary information from the regions of interest, SemCom can send only the semantic information to generate the desired DT. One of the key features of this system is the integration of XAI through the GradCAM technique. GradCAM computes the derivatives of the output classes with respect to the feature maps, creating heat maps that show exactly which parts of the image contributed the most to the detection. For each camera stream, a heat map is generated and superimposed over the original image, highlighting areas of high interest with hot colors, as shown in Figure 19, which shows three frames, one from each camera, with the original image displayed on the left and the right image containing the heatmaps. These images allow the users to verify that the model is observing critical pedestrian features from different directions. The heatmaps act as a window into the "logic" of the YOLO model, revealing that the model correctly focuses on pedestrians, omitting irrelevant background information.

After the detection and visualization stages, the system computes the corresponding semantic data, and then the SemCom model sends this information over the V2x networks in the form of lightweight data packets. This SemCom approach transmits only the required data, allowing maximum transmission speed and optimal system performance. On the receiver side, semantic information is used to generate a dynamic DT, capable of being generated in real time, offering the analogous advantages of a live DT, such as real-time accident alert. In addition, GradCAM visualizations offer a secondary but useful function, providing a means of continuous verification and debugging. By observing the heatmaps from each camera, users can ensure that the YOLO model focuses on the expected characteristics, increasing confidence in the system and allowing for continuous improvement.

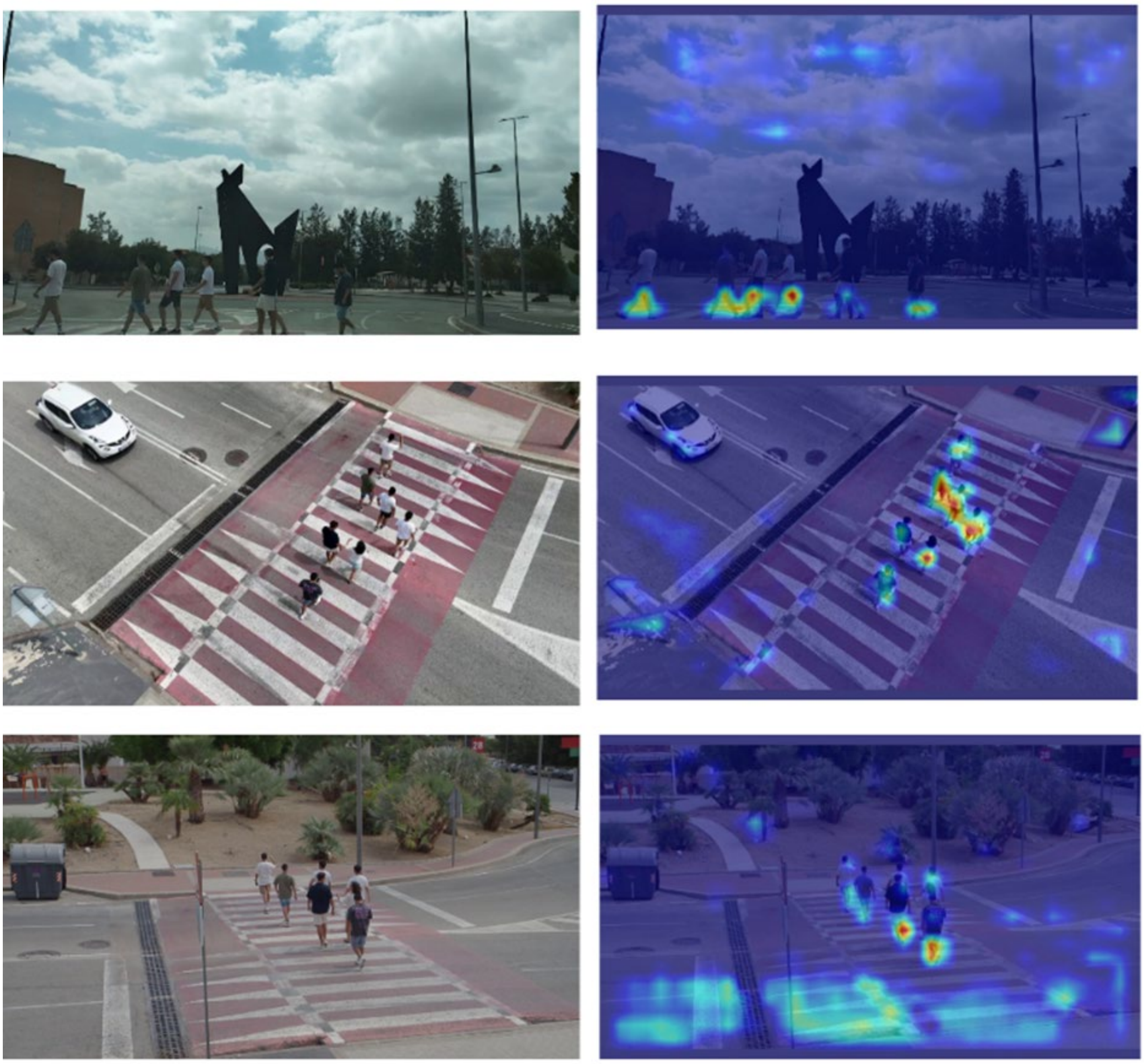


Figure 19: GradCAM heatmaps for each camera of the multi-view V2x scenario.

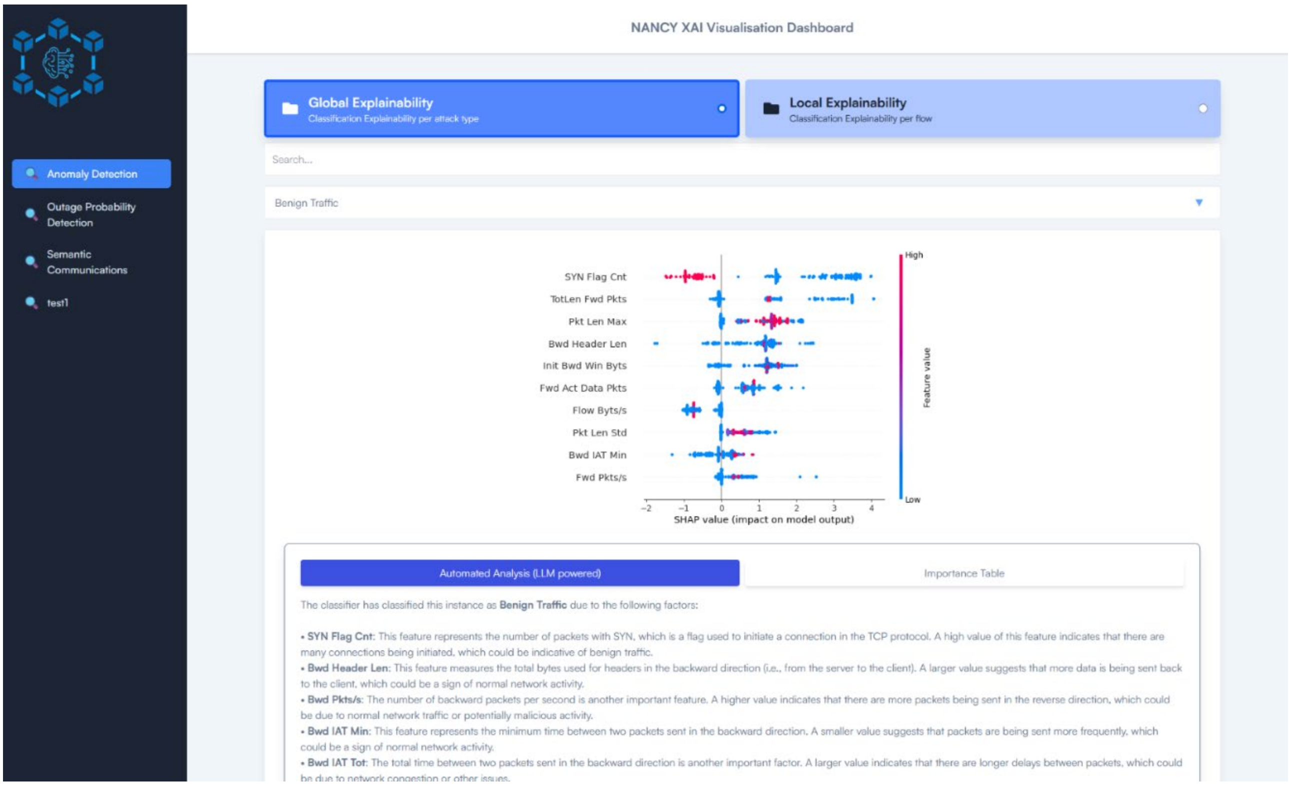
4. XAI Dashboard Implementation

The NANCY XAI Dashboard is a web-based application with the objective of providing a comprehensive and interactive platform for visualizing and analyzing Explainable AI (XAI) outputs. It is a meeting point for network operators and other stakeholders to understand the AI model decision-making processes that are applied in network management. The dashboard is built to handle image data in conjunction with associated JSON metadata so that users can naturally navigate, filter, and analyze AI model outputs along with their explanations.

4.1. Dashboard Architecture

The NANCY XAI Dashboard provides a scalable environment for the visualization and analysis of explainable AI outputs. The core design is based on a three-panel responsive layout, namely the "Anomaly Detection", the "Outage Probability Detection" and the "Semantic Communications". All panels with their sub-windows and drop-down menus are showcased in the Annex. The landing page is demonstrated below in Figure 20:

- Navigation Sidebar: Persistent icon-driven navigation to quickly select analytical modules.
- Upper Content Panel: A tab-based setting enabling easy toggling between Global and Local explainability analyses.
- Dynamic Visualization Panel: Interactive data visualizations and analytical results are dynamically rendered.



The classifier has classified this instance as **Benign Traffic** due to the following factors:

- **SYN Flag Cnt**: This feature represents the number of packets with SYN, which is a flag used to initiate a connection in the TCP protocol. A high value of this feature indicates that there are many connections being initiated, which could be indicative of benign traffic.
- **Bwd Header Len**: This feature measures the total bytes used for headers in the backward direction (i.e., from the server to the client). A larger value suggests that more data is being sent back to the client, which could be a sign of normal network activity.
- **Bwd Pkts/s**: The number of backward packets per second is another important feature. A higher value indicates that there are more packets being sent in the reverse direction, which could be due to normal network traffic or potentially malicious activity.
- **Bwd IAT Min**: This feature represents the minimum time between two packets sent in the backward direction. A smaller value suggests that packets are being sent more frequently, which could be a sign of normal network activity.
- **Bwd IAT Tot**: The total time between two packets sent in the backward direction is another important factor. A larger value indicates that there are longer delays between packets, which could be due to network congestion or other issues.

Figure 20: Dashboard overview

Using the Next.js 14 App Router, the dashboard integrates a file-system-based routing (/app/dashboard/[foldername]/). It provides easy adoption of new tabs and significantly boosts the scalability of an application. The use of abstract components assures universal layout as well as smooth integration, excluding duplication of code, and making processes faster to implement. Table 4 indicates the dashboard's main components.

Table 4: Dashboard's main components

Component Path	Functionality and Purpose
/app/dashboard	Manages overall layout and ensures consistency across pages
/app/dashboard/[tab]	Enables scalable and dynamic integration of analytical modules
Abstract Components	Template layouts ensuring visual consistency and rapid integration

4.2. LLM Integration for Results Explanation

In cybersecurity threat detection, explainability is crucial for understanding and validating AI-driven decisions. Traditional machine learning classifiers often operate as "black boxes," making it challenging for security analysts to interpret why a specific event is flagged as malicious. To address this, we have integrated a Large Language Model (LLM)-Powered Analysis Component that leverages SHAP values [28] to provide human-readable explanations for classifier decisions.

By incorporating an LLM-driven approach, we aim to enhance transparency, trust, and usability in our security intelligence framework, ensuring that AI-generated insights are both actionable and understandable.

An LLM is a deep learning-based artificial intelligence system that is trained on vast amounts of textual data to understand, generate, and analyze human-like text. LLMs utilize transformer-based architectures [29] to process and generate natural language with a high degree of fluency and contextual awareness. These models are capable of performing a wide range of tasks, including text summarization, machine translation, question answering, and content generation.

One of the core capabilities of LLMs is contextual understanding and decision-making, and therefore, they have use cases in healthcare, finance, law, and customer service. In healthcare, for example, LLMs can assist in clinical decision-making by analyzing patient records and by providing diagnostic suggestions. In finance, LLMs can study financial reports, detect anomalies in transactions, and build market research.

One of the main uses of LLMs is in XAI, where they are used to explain intricate machine learning models by transforming quantitative feature attributions—like SHAP values—into actionable, human-understandable insights. In this way, LLMs allow users to see the rationale underlying AI-generated predictions, making machine learning models more interpretable and accessible to non-experts.

For our LLM-powered analysis component, we utilize the Mistral-7B-cybersecurity-rules[1] model. This model is specifically fine-tuned for cybersecurity rule generation and explainability.

- Base Model: Mistral-7B-Instruct-v0.2
- Specialization: Threat detection and rule generation

[1] https://huggingface.co/jcordon5/Mistral-7B-cybersecurity-rules

- Training Data: A curated corpus of 950 cybersecurity rules from SIGMA, YARA, and Suricata[2] repositories
- Primary Use Case: Automated rule creation, security event analysis, and SHAP value interpretation

The model's cybersecurity-specific fine-tuning makes it an ideal candidate for explaining classifier decisions, as it understands the relationships between threat indicators and detection rules.

The core technologies utilized for the project implementation are:

- Large Language Model: Mistral-7B-cybersecurity-rules (fine-tuned Mistral-7B-Instruct-v0.2)
- Framework: Hugging Face Transformers
- Runtime Environment: Python 3.12.5[3] with PyTorch[4]
- Hardware Acceleration: CUDA-compatible for GPU acceleration

The main.py script serves as the primary entry point for the analysis workflow and implements the following functionalities:

- Model Loading and Configuration:
  - Loads the Mistral-7B-cybersecurity-rules model from Hugging Face
  - Configures tokenizer settings with appropriate padding token handling
  - Sets maximum input token length constraint
- Analysis Workflow:
  - Processes SHAP values from explainable AI components
  - Constructs appropriate prompts for the LLM with feature importance data
  - Handles the generation of explanatory text through the Transformers pipeline
- Output Processing:
  - Formats and presents the model's explanation in a user-friendly manner
  - Highlights key features using markdown syntax for improved readability
  - Organizes explanations into overview, bullet points, and conclusion sections

Table 5: Dependencies for the LLM implementation

Libary	Version	Description
torch[5]	2.5.1	Supplies the computational backend for model operations
accelerate[6]	1.0.1	Optimizes model loading and inference for better performance
peft[7]	0.13.2	Supports parameter-efficient fine-tuning techniques if model asjustments are needed
huggingface-hub[8]	0.26.2	Facilitates model downloading and version management
nvidia-cuda[9]	-	Enable GPU acceleration for faster inference

[2] https://suricata.io
[3] https://www.python.org/downloads/release/python-3125/
[4] https://pytorch.org
[5] https://pytorch.org/get-started/previous-versions/
[6] https://pypi.org/project/accelerate/
[7] https://github.com/huggingface/peft
[8] https://huggingface.co/docs/hub/index
[9] https://developer.nvidia.com/cuda-toolkit

transformers[10]	4.46.1	Provides the core functionality for loading and running the LLM

The component accepts the following input data:

- Class prediction (e.g., "Outage_Risk")
- Feature importance rankings with associated descriptions
- Feature importance scores from SHAP analysis

The following optimization techniques enhance AI model performance by improving efficiency, leveraging hardware acceleration, and managing resource constraints effectively.

- Torch bfloat16 Precision: Reduces memory footprint while maintaining computational accuracy
- Device Mapping: Automatically utilizes available GPU resources for accelerated inference
- Token Length Management: Enforces maximum token constraints to prevent resource exhaustion

The explanation output of the LLM follows a structured format:

- Brief Overview: Summarizes why the model made its classification
- Feature Analysis: Bullet-pointed explanations of each feature's relevance
- Conclusion: Synthesizes the feature analysis into a coherent final assessment

The process, illustrated in Figure 12, begins with the XAI Component, which generates SHAP values (both for Outage prediction as well as Anomaly Detection) to quantify the contribution of individual features in a model's decision-making. These values are then processed through an LLM-powered Analysis module, which interprets the SHAP values and generates human-readable explanations. Finally, the processed explanations are sent to the NANCY Explainable AI Dashboard, which provides a user-friendly interface for understanding and visualizing model behavior. This workflow enhances transparency and interpretability in AI-driven decision systems.

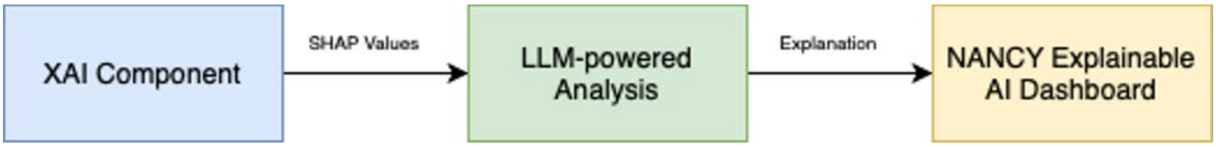


Figure 21: Data flow diagram between the XAI component, LLM-Powered Analysis component and NANCY Explainable AI Dashboard

The component's output format is designed to integrate seamlessly with the NANCY XAI Dashboard, providing:

- Markdown-formatted text suitable for web display
- Bold highlighting of feature names for improved readability
- Bullet-point organization for structured presentation

Future development efforts will focus on enhancing explainability, expanding model support, optimizing performance, and improving efficiency through batch processing.

[10] https://pypi.org/project/transformers

- Explanation Customization: Implementing options for different explanation styles and detail levels
- Model Alternatives: Support for additional LLMs beyond the current Mistral-7B implementation
- Performance Optimization: Further tuning for reduced latency and memory usage
- Batch Processing: Adding capabilities to analyze multiple predictions in a single run

The integration of the LLM-Powered Analysis Component successfully enhances explainability by translating SHAP values into interpretable insights. To achieve this, the system utilizes a structured prompt to guide the LLM in producing clear, human-readable explanations for classification results. Below is the exact prompt used to generate the explanation:

```
messages = [{
   "role": "user",
   "content": '''You are a cybersecurity expert, and you are responsible to analyze the SHAP values
produced by an explainable AI component.
             You are responsible to analyze why the classifier has classified the instance to the given class.
             I want at first a brief overview, then the explanation in bullet points (the bullet points should
be displayed as •) which will include markdown, and finally a conclusion.
             The brief overview should begin with the following based on the input of SHAP values for
the given class.
             I want to explain whether the given features really matter in predicting the associated class.
             I do not want to take into consideration the importance of each feature. I do not want any
introductory information and any headings e.g. Summary.
             Make bold all the features/classes and do not use any " in the response.
             Here is the output from the classifier.

            {
   "class": "Outage_Risk",
   "top_features": [
      {
         "feature_name": "tx_pkts",
         "importance": 2.8807356357574463,
         "description": "Number of transmitted packets in downlink"
      },
      {
         "feature_name": "dl_cqi",
         "importance": 0.8394917249679565,
         "description": "Channel Quality Indicator for downlink"
      },
      {
         "feature_name": "sum_granted_prbs",
         "importance": 0.47345030307769775,
         "description": "Sum of granted Physical Resource Blocks"
      },
      {
         "feature_name": "sum_requested_prbs",
         "importance": 0.25738516449928284,
         "description": "Sum of requested Physical Resource Blocks"
```

```
    },
    {
      "feature_name": "dl_buffer",
      "importance": 0.03884560987353325,
      "description": "Buffer size in the downlink direction"
    }
  ]
}
Give an explanation to a beginner without referencing the SHAP values.'''
}]
```

At the core of this system is a carefully crafted prompt that ensures the LLM generates explanations that are concise, informative, and tailored for a non-expert audience. The prompt follows a structured format designed to guide the model in producing clear and contextually relevant insights. First, the LLM is assigned a specific role, acting as a cybersecurity expert responsible for analyzing SHAP values produced by an explainable AI component. This role assignment helps the model maintain a focused and domain-specific approach while generating explanations.

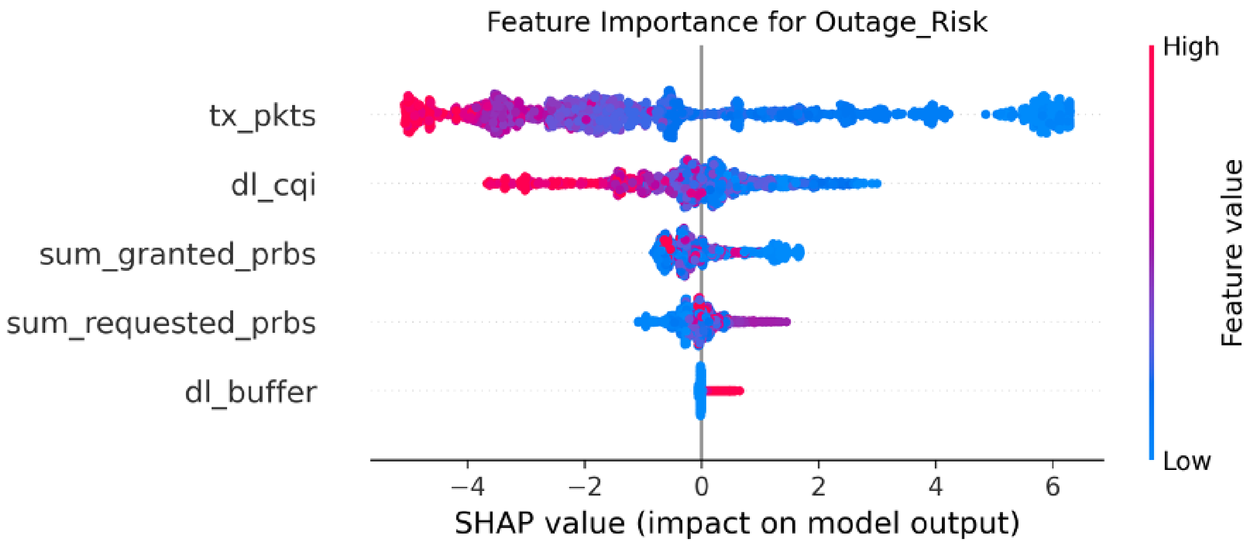


Figure 22: SHAP values example for the Outage Risk prediction

The task definition steers the LLM to explain why a prediction was made without explicitly mentioning SHAP values. Instead of providing comments on numerical feature attributions, the model focuses on interpreting these results in natural language such that the results are more user-friendly. The output format is also designed in the same way such that each response has a summary of the outcome of the classification, along with a detailed description of the predominant features that influenced the decision. The descriptions are laid out to be easily readable and highlight how particular features contributed to the model's decision to achieve the forecast class. The answer is concluded with a summary correlating the recognized features with the classification outcome, reiterating the justification for the model's decision.

To maintain consistency and readability, specific formatting rules are enforced within the prompt. All feature names and class labels are presented in bold text to ensure clarity and emphasis. Additionally, the explanation follows a structured flow without introductory headings like "Summary" or extraneous contextual information beyond the requested analysis. These formatting constraints help create a streamlined and professional output that aligns with the needs of analysts and decision-makers.

By following these standards, the LLM ensures that complex AI-generated outputs are presented in a way that is easy to understand. This structured approach reduces the cognitive load on analysts, enabling them to quickly understand the reasoning behind AI-generated decisions. Also, it enhances model transparency, which enhances the trustworthiness and reliability of AI systems in high-risk scenarios where interpretability is paramount.

Below is an example of the system's output when analyzing a network activity instance:

```
{
  "class": "Outage Risk",
  "analysis": "The classifier has classified this instance as **Outage Risk** due to the following factors:\n\n• **tx_pkts**: This feature represents the number of transmitted packets in the downlink direction. It is important because it indicates the amount of data being sent from the network to the user's device. A higher value of this feature could indicate a higher risk of network congestion or outage.\n• **dl_cqi**: This feature stands for Channel Quality Indicator for downlink. It measures the quality of the downlink channel, which is essential for understanding the user's experience with the network. A lower value of this feature could indicate poor signal strength, leading to a higher risk of network congestion or outage.\n• **sum_granted_prbs**: This feature represents the sum of granted Physical Resource Blocks in the downlink direction. It is important because it indicates the amount of resources allocated to the user's device for data transmission. A higher value of this feature could indicate a higher risk of network congestion or outage due to the increased demand for resources.\n• **sum_requested_prbs**: This feature represents the sum of requested Physical Resource Blocks in the downlink direction. It is important because it indicates the user's device's demand for resources, which can affect the network's performance and availability. A higher value of this feature could indicate a higher risk of network congestion or outage due to the increased demand for resources.\n• **dl_buffer**: This feature represents the buffer size in the downlink direction. It is important because it indicates the amount of data that can be stored in the network's buffer before being transmitted to the user's device. A lower value of this feature could indicate a higher risk of network congestion or outage due to the inability to store and transmit data efficiently.\n\nIn conclusion, these features are crucial for predicting the associated class because they provide insights into the network's performance and availability. By understanding the relationship between these features and the classifier's output, we can identify potential issues that may lead to network congestion or outages. This knowledge helps us to proactively manage network resources and improve user experience."
}
```

The system was evaluated based on its ability to generate accurate, relevant, and interpretable explanations. The results confirm that:

- The LLM correctly associates features with cybersecurity-relevant attributes, ensuring accurate SHAP-based explanations.
- The generated insights align with domain knowledge, reinforcing trust in the AI-driven decision-making process.
- The model significantly reduces the cognitive load on security analysts by automating the interpretation of complex ML outputs.

4.3. Visualization Components and User Interface

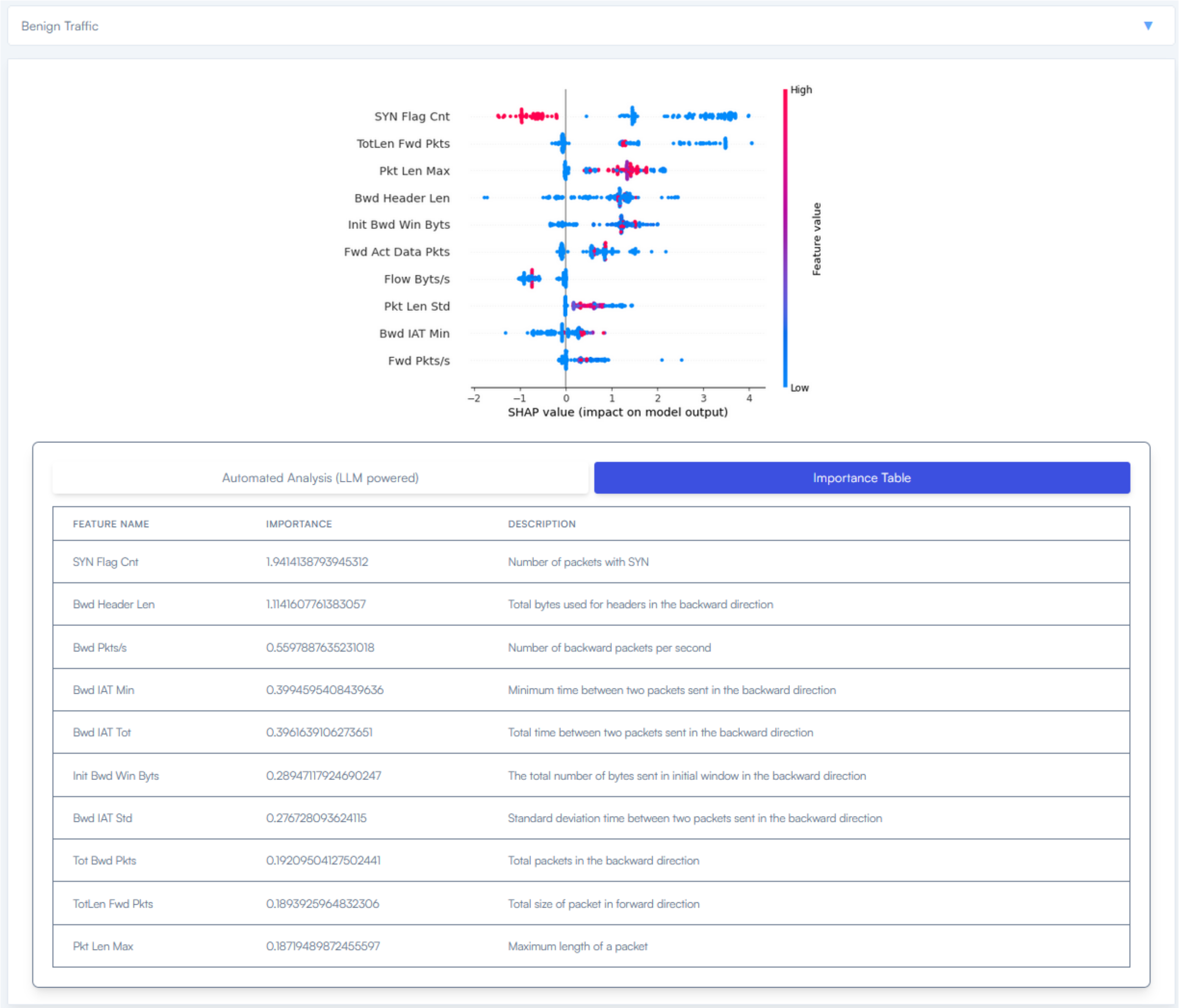
Visualization components in the dashboard are reusable, modular, and designed for optimal user interaction, as shown in Figure 23:

- Interactive Dot Plots: Offer compact visual rendition of global feature impacts.
- Data Grids: Offer detailed numerical analysis, with filtering and sorting capabilities.
- Statistical Cards: Offer essential metrics concisely and efficiently.

The user interface is designed to favor an intuitive look and feel and user experience:

- Progressive Disclosure: Discloses detailed analytical information progressively to prevent cognitive overload.
- Responsive Interactivity: Anticipates instant responses to improve interaction with the user and ensure transparency in navigation.

Centralized configuration supports theming and internationalization, which ensures the dashboard's flexibility to cope with different user requirements is simple and seamless.



FEATURE NAME	IMPORTANCE	DESCRIPTION
SYN Flag Cnt	1.9414138793945312	Number of packets with SYN
Bwd Header Len	1.1141607761383057	Total bytes used for headers in the backward direction
Bwd Pkts/s	0.5597887635231018	Number of backward packets per second
Bwd IAT Min	0.3994595408439636	Minimum time between two packets sent in the backward direction
Bwd IAT Tot	0.3961639106273651	Total time between two packets sent in the backward direction
Init Bwd Win Byts	0.28947117924690247	The total number of bytes sent in initial window in the backward direction
Bwd IAT Std	0.276728093624115	Standard deviation time between two packets sent in the backward direction
Tot Bwd Pkts	0.19209504127502441	Total packets in the backward direction
TotLen Fwd Pkts	0.1893925964832306	Total size of packet in forward direction
Pkt Len Max	0.18719489872455597	Maximum length of a packet

Figure 23: Front-end Importance Table

Modularity allows for rapid updates and extension, supporting the addition of new visualization features with minimal effort. Table 6 lists the functional components of the developed Dashboard.

Table 6: Dashboard's functional components

Component / UI Aspect	Description and Benefit	Application Scenario
Dot Plots	Visualize global feature importance clearly	Global model interpretability
Data Grids	Detailed interactive numerical analysis	In-depth numerical data analysis
Statistical Cards	Quick summary of key metrics	Rapid analytical insights
Progressive Disclosure	Gradually reveals detailed data	Enhanced usability
Consistent Visual Style	Cohesive user-friendly design	Improved user comfort
Responsive Interaction	Immediate UI response	Active user engagement

4.4. Scalability and Extensibility

The dashboard becomes scalable and extendable by applying architectural patterns:

- Dynamic Routing: Enables immediate addition of fresh analytical views.
- Abstract and Reusable Components: Enable consistent and rapid development throughout the dashboard.
- Independent Module Integration: Enable components to be reused outside the context of the dashboard, allowing separate applications.

Table 7 indicates the architectural features of the developed dashboard, that enables its scalability:

Table 7: Dashboard's architectural features

Feature	Implementation Approach	Operational Advantage
Dynamic Routing	Next.js folder-based dynamic routes	Rapid and seamless expansion
Abstract Components	Reusable UI templates	Reduced coding effort and consistency
Independent Modules	Modular standalone usage	Flexible reuse and easy deployment

5. Integration and Deployment

Effective integration and deployment plans are critical to ensuring that the NANCY XAI Dashboard runs stably and scales well. The below explains the strategies employed to streamline component communications, optimize performance, and guarantee good application stability.

5.1. Component Integration

The integration strategy of the NANCY XAI Dashboard enables seamless communication between frontend and backend data services through properly structured, well-defined processes. Well-designed API endpoints created following Next.js 14 App Router conventions (naming convention: route.ts), handle HTTP requests nicely through properly structured methods such as GET and POST. This kind of well-organized backend-to-frontend communication simplifies data transfer for efficient fetching and transformation of complex data structures, particularly JSON metadata. Table 8 lists the technologies used for developing the dashboard.

Table 8: Dashboard's Technology Stack

Technology	Version	Purpose
Next.js[11]	14.2.24	Frontend routing and server-side rendering
React[12]	18.3.1	UI library for building interactive interfaces
TypeScript[13]	5.7.3	Static typing to enhance code quality
Tailwind CSS[14]	3.4.17	Utility-first CSS framework for responsive design
React Hooks[15]	useState/useEffect	State management for efficient rendering

The modular component-based architecture makes integration possible by means of clean separation of concerns. The layout components provide structure consistency, and feature-specific components encapsulate complex logic independently. Common UI components (e.g., loaders, cards, data displays) are utilized as reusable components. Hierarchical layering of the components enables independent development, testing, and deployment, making collaboration efficient and CI/CD processes seamless. Table 9 shows some of the technical details of the dashboard.

Table 9: Dashboard's technical details

Integration Aspect	Technical Details	Advantage
API Endpoint Management	Structured Next.js App Router API routes	Efficient, clear communication
JSON Data Handling	Structured parsing and dynamic data formatting	Accurate data visualization, responsiveness
Component-Based Integration	Hierarchical modularity and encapsulated logic	Independent deployment, streamlined CI/CD

[11] https://nextjs.org/
[12] https://react.dev/
[13] https://www.typescriptlang.org/
[14] https://tailwindcss.com/
[15] https://react.dev/reference/react/hooks

5.2. Performance Optimization

Optimization methods of the dashboard are specifically engineered to handle dynamic user interactions and long data operations in the best possible way. React hooks (useState, useEffect) offer state management that is responsive, reducing unnecessary rendering loops and enhancing user interactions. In addition, the application employs lazy loading techniques for components and data resources alike so that resources are loaded precisely when they are needed, thereby significantly improving initial load times and decreasing memory usage.

Loading states are managed strategically to provide immediate visual feedback while fetching data, making the process transparent and involving the user. Ongoing monitoring and profiling guarantee that bottlenecks in the performance can be immediately identified and resolved to maintain ongoing high levels of performance. Table 10 summarizes the performance strategies that are applied on the dashboard.

Table 10: Dashboard's performance strategies

Performance Strategy	Implementation Method	Operational Benefit
React Hooks for State Management	Efficient localized state management	Enhanced responsiveness, reduced rendering time
Lazy Loading	Dynamic importing of components/data	Improved initial load, reduced memory footprint
Component Isolation	Modular architecture, isolated state	Stable performance, reduced risk of regressions
Immediate Loading Feedback	Visual feedback during data fetch operations	Improved user experience, increased transparency
Continuous Monitoring	Regular performance profiling and optimization	Consistent performance, proactive issue handling

6. Conclusion and Outlook

The development and integration of the XAI toolbox and functionalities are a significant milestone for the whole NANCY platform. Through the XAI toolbox, the concepts of Familiarity, Knowledgeability, and Fairness are integrated. The deliverable demonstrates the technical feasibility of integrating explainability mechanisms directly into the AI pipelines used in network management.

Both global and local explanation techniques using various XAI techniques, tailored for network traffic classification and outage prediction and semantic communication, are developed. This dual-layer approach ensures that stakeholders can understand not only general model behavior but also case-specific decisions. The NANCY XAI Dashboard provides a scalable and user-friendly environment for the visualization and analysis of explainable AI outputs.

The LLM-Powered Analysis Component successfully converts numerical attributions into insightful cybersecurity information, bridging the gap between AI-driven threat detection and human interpretability. This feature strengthens the trust and usability of AI-based cybersecurity solutions, thus improving threat analysis and response efficiency.

The toolbox and dashboard were designed to be extensible, using modular and scalable software patterns that can be easily adapted to new network management scenarios. NANCY XAI toolbox functionalities support WP5 goals. Additionally, it enhances the usability of AI components, which are aligned with the requirements derived from WP3 and WP4. The XAI toolbox will be validated in Greek in-lab testbed, under WP6, according to D6.2 “NANCY Integrated system – Initial version”, in the upcoming months.

References

[1] V. Sahu, N. Sahu, and R. Sahu, “Challenges and Opportunities of 5G Network: A Review of Research and Development,” AJECE, vol. 8, no. 1, pp. 11–20, Jul. 2024

[2] Steve Moyle, Andrew Martin, Nicholas Allott, "XAI Human-Machine collaboration applied to network security," Frontiers in Computer Science, vol. 6, May 2024.

[3] A. Adadi and M. Berrada, "Peeking Inside the Black-Box: A Survey on Explainable Artificial Intelligence (XAI)," IEEE Access, vol. 6, pp. 52138-52160, 2018.

[4] G. Rjoub et al., "A Survey on Explainable Artificial Intelligence for Cybersecurity," IEEE Transactions on Network and Service Management, vol. 20, no. 4, pp. 5115-5140, Dec. 2023.

[5] European Union, "Article 22 GDPR. Automated individual decision-making, including profiling," [Online]. Available: https://gdpr-text.com/read/article-22/

[6] M. Saarela and V. Podgorelec, “Recent Applications of Explainable AI (XAI): A Systematic Literature Review,” Applied Sciences, vol. 14, no. 19, p. 8884, Oct. 2024.

[7] S. Ness, V. Eswarakrishnan, H. Sridharan, V. Shinde, N. Venkata Prasad Janapareddy and V. Dhanawat, "Anomaly Detection in Network Traffic Using Advanced Machine Learning Techniques," IEEE Access, vol. 13, pp. 16133-16149, 2025.

[8] S. M. Lundberg, S.-I. Lee, "A unified approach to interpreting model predictions," 31st International Conference on Neural Information Processing Systems, pp. 4765–4774, 2017.

[9] A. M. Aljohani, I. Elgendi, "The interest of hybridizing explainable AI with RNN to resolve DDoS attacks: A comprehensive practical study," International Journal of Network Security & Its Applications, 2024.

[10] M. T. Ribeiro, S. Singh, C. Guestrin, "Why should I trust you?: Explaining the predictions of any classifier," *22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining,* p. 1135–1144, 2016.

[11] Maryam Daud, Ghalib Shah, Kausar Parveen, Ashar Ahmed Fazal, "A Deep Intelligent Hybrid Intrusion Detection Framework with LIME Explainability for Fog-Based IoT Networks (DIHIF-LIME)," *Kurdish Studies, vol. 12, no. 5,* p. 484-492, 2024.

[12] S. Ahmed, M. S. Kaiser, M. Shahadat Hossain, K. Andersson, "A Comparative Analysis of LIME and SHAP Interpreters With Explainable ML-Based Diabetes Predictions," *IEEE Access, vol 13,* pp. 37370-37388, 2025.

[13] C. De Alwis, A. Kalla, Q.-V. Pham, P. Kumar, K. Dev, W.-J. Hwang, and M. Liyanage, "Survey on 6G Frontiers: Trends, Applications, Requirements, Technologies and Future Research," IEEE Open Journal of the Communications Society, vol. 2, pp. 836-886, 202.

[14] W. Saad, M. Bennis, and M. Chen, “A vision of 6G wireless systems: Applications, trends, technologies, and open research problems”, IEEE Network, vol. 34, no. 3, pp. 134–142, Oct. 2019.

[15] X. Wang, D. Ye, C. Feng, H. H. Yang, X. Chen, and T. Q. S. Quek, "Trustworthy Image Semantic Communication with GenAI: Explainablity, Controllability, and Efficiency.", *arXiv preprint arXiv:2408.03806,* 2024.

[16] C. Liu, C. Guo, Y. Yang, W. Ni, Y. Zhou, L. Li, "Explainable Semantic Communication for Text Tasks," IEEE Internet of Things Journal, vol. 11, no. 24, pp. 39820-39833, Dec. 2024.

[17] S. Ma, W. Qiao, Y. Wu, H. Li, G. Shi, and D. Gao, "Task-Oriented Explainable Semantic Communications," in IEEE Transactions on Wireless Communications, vol. 22, no. 12, pp. 9248-9262, Dec. 2023.

[18] S. Jiang et al., "Reliable Semantic Communication System Enabled by Knowledge Graph," Entropy, vol. 24, no. 6, p. 846, Jun. 2022

[19] L. Hu, Y. Li, H. Zhang, L. Yuan, F. Zhou, and Q. Wu, "Robust semantic communication driven by knowledge graph," *9th International Conference on Internet of Things: Systems, Management and Security (IOTSMS), 2022, p*p. 1–5.

[20] Y. Wang, M. Chen, W. Saad, T. Luo, S. Cui and H. V. Poor, "Performance Optimization for Semantic Communications: An Attention-based Learning Approach," IEEE Global Communications Conference (GLOBECOM), 2021, pp. 1-6.

[21] S. Sun, Z. Qin, H. Xie and X. Tao, "Task-Oriented Explainable Semantic Communications Based on Structured Scene Graphs," IEEE Global Communications Conference (GLOBECOM), 2023, pp. 3222-3227.

[22] V. Kouvakis, S. E. Trevlakis and A. - A. A. Boulogeorgos,, "Semantic Communications for Image-Based Sign Language Transmission," *IEEE Open Journal of the Communications Society,* vol. 5, p. 1088-1100, 2024.

[23] "Colosseum O-RAN ColORAN Dataset," [Online]. Available: https://github.com/wineslab/colosseum-oran-coloran-dataset.

[24] S. E. Trevlakis, N. Pappas and A. - A. A. Boulogeorgos, "Toward Natively Intelligent Semantic Communications and Networking,", *IEEE Open Journal of the Communications Society,* vol. 5, p. 1486-1503, 2024.

[25] V. Kouvakis, S. E. Trevlakis, A. - A. A. Boulogeorgos, T. Tsiftsis, K. Singh, and N. Qi, "When Sign Language Meets Semantic Communications," *IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, Valencia, Spain, 2024.

[26] Kouvakis, V., Mitsiou, L., Trevlakis, S. E., Boulogeorgos, A.- A. A., & Tsiftsis, T., "American Sign Language dataset for semantic communications [Data set]" IEEEDataPort & Zenodo, 2025.

[27] Ramon Sanchez-Iborra, Rodrigo Asensio-Garriga, Gonzalo Alarcon-Hellin, Luis Bernal-Escobedo, Stylianos Trevlakis, Theodoros Tsiftsis, Antonio Skarmeta, "Multi-perspective Traffic Video Recording", IEEE Dataport, January 20, 2025.

[28] S. Lundberg, S. Lee, "A unified approach to interpreting model predictions," 31st International Conference on Neural Information Processing Systems, 2017, pp. 4768–4777.

[29] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. Gomez, L. Kaiser, and I. Polosukhin, "Attention is All you Need," Advances in Neural Information Processing Systems, vol. 30, pp. 5998 - 6008, 2017.

Annex – Screenshots from the Graphical User Interface

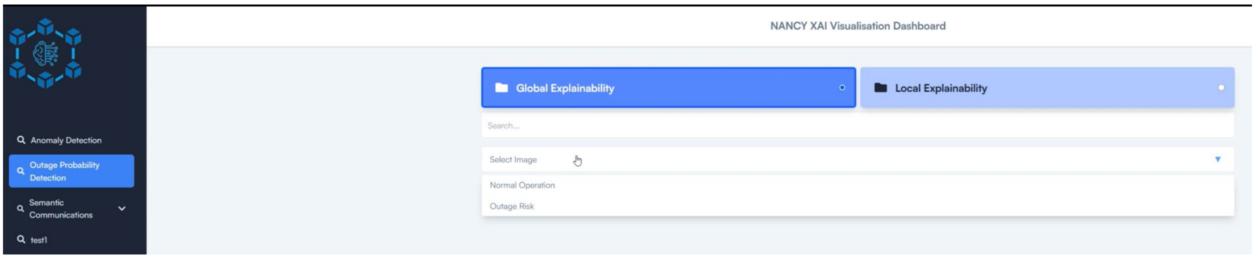


Figure A 1: Drop-down menu - Global explainability

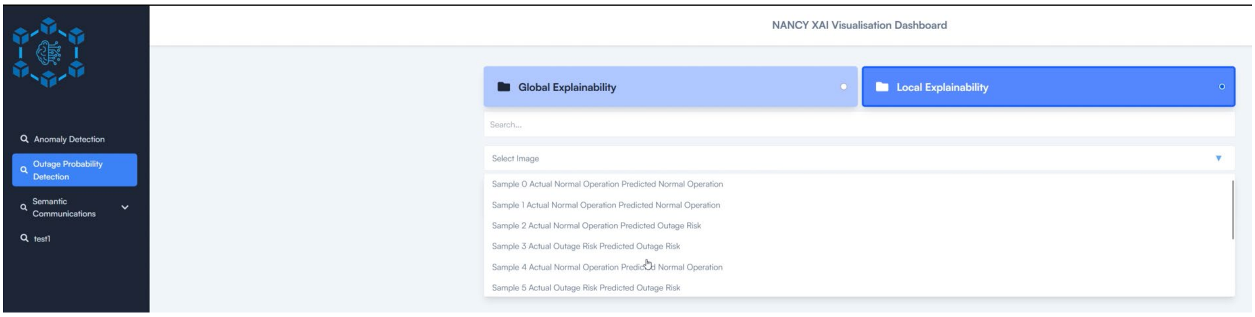

NANCY XAI Visualisation Dashboard
Anomaly Detection
Outage Probability Detection
Semantic Communications
test1
Global Explainability
Local Explainability
Search...
Select Image
Sample 0 Actual Normal Operation Predicted Normal Operation
Sample 1 Actual Normal Operation Predicted Normal Operation
Sample 2 Actual Normal Operation Predicted Outage Risk
Sample 3 Actual Outage Risk Predicted Outage Risk
Sample 4 Actual Normal Operation Predicted Normal Operation
Sample 5 Actual Outage Risk Predicted Outage Risk

Figure A 2: Drop-down menu - Local explainability

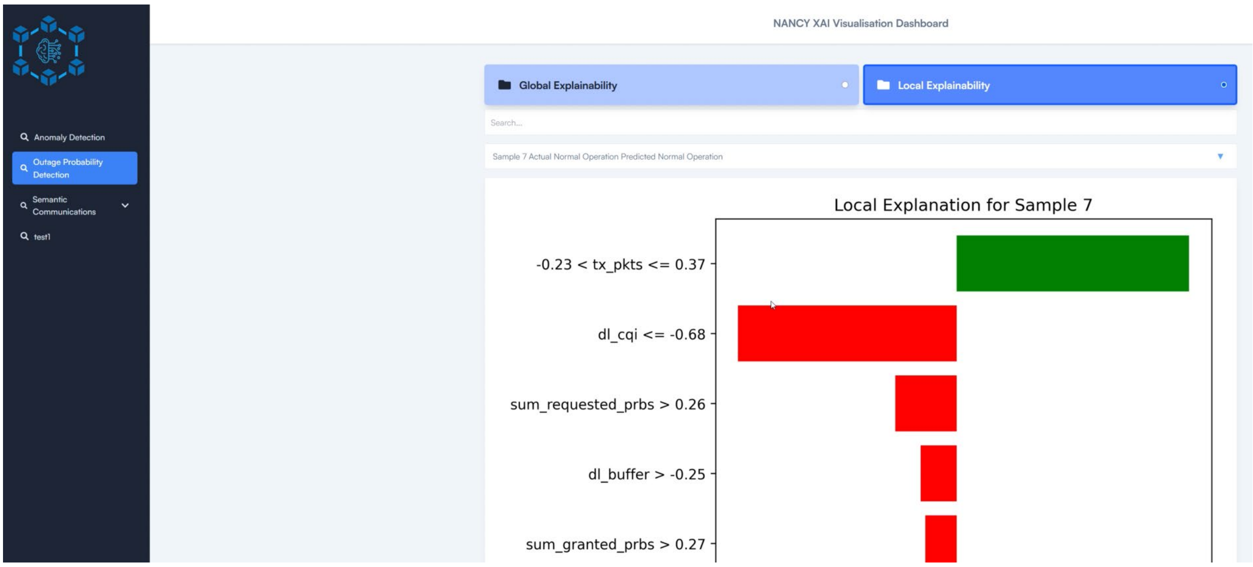

NANCY XAI Visualisation Dashboard
Anomaly Detection
Outage Probability Detection
Semantic Communications
test1
Global Explainability
Local Explainability
Search...
Sample 7 Actual Normal Operation Predicted Normal Operation
Local Explanation for Sample 7
-0.23 < tx_pkts <= 0.37
dl_cqi <= -0.68
sum_requested_prbs > 0.26
dl_buffer > -0.25
sum_granted_prbs > 0.27

Figure A 3: Local explainability sample

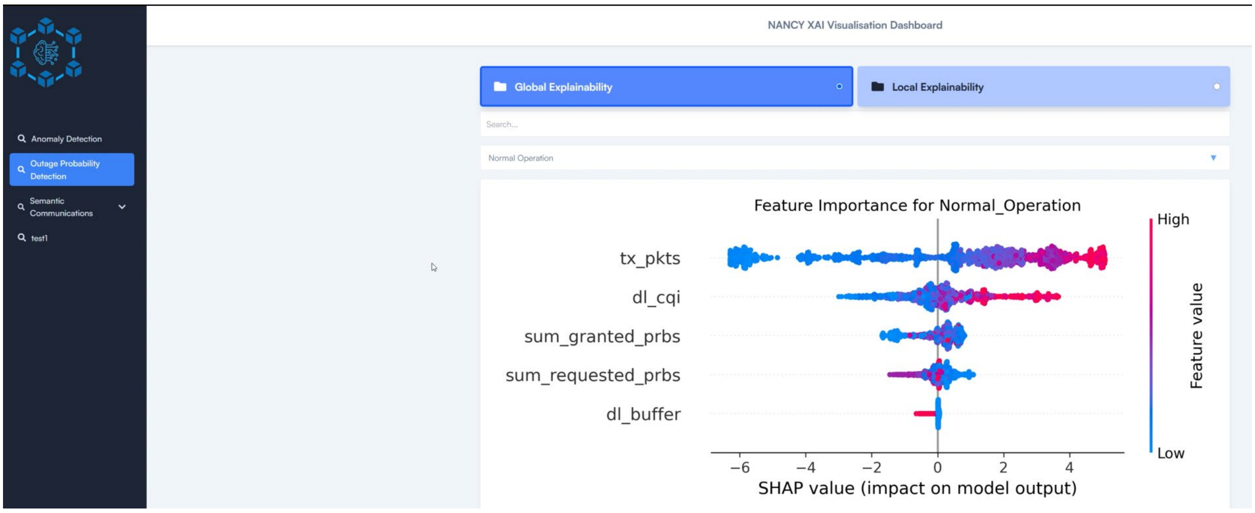


Figure A 4: Global explainability sample

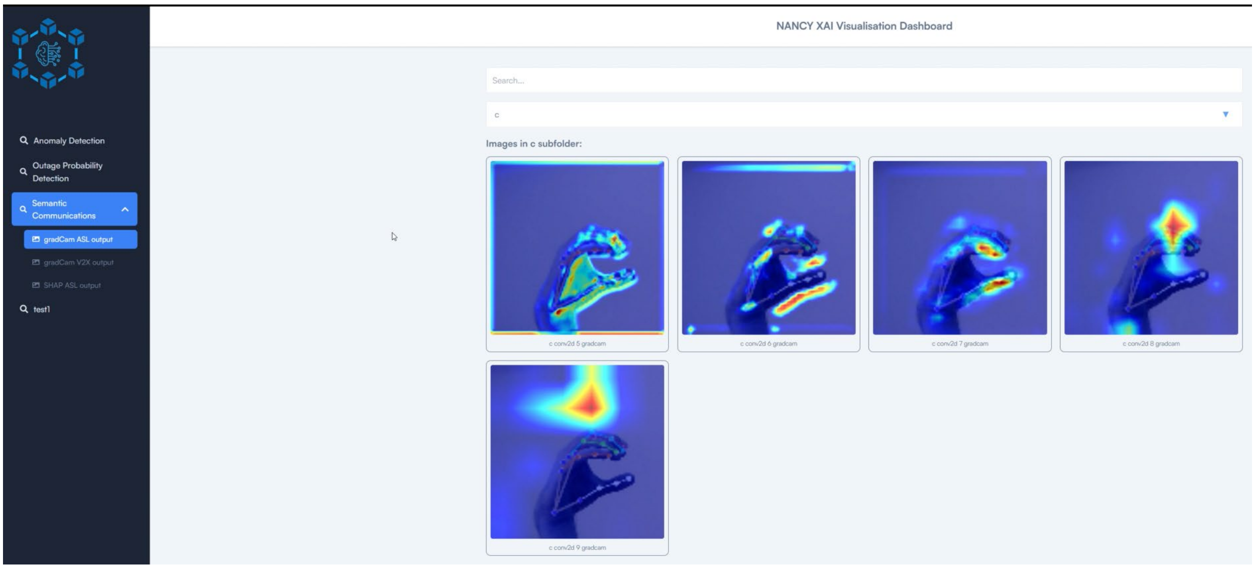


Figure A 5: Semantic communications menu – gradCam ASL output

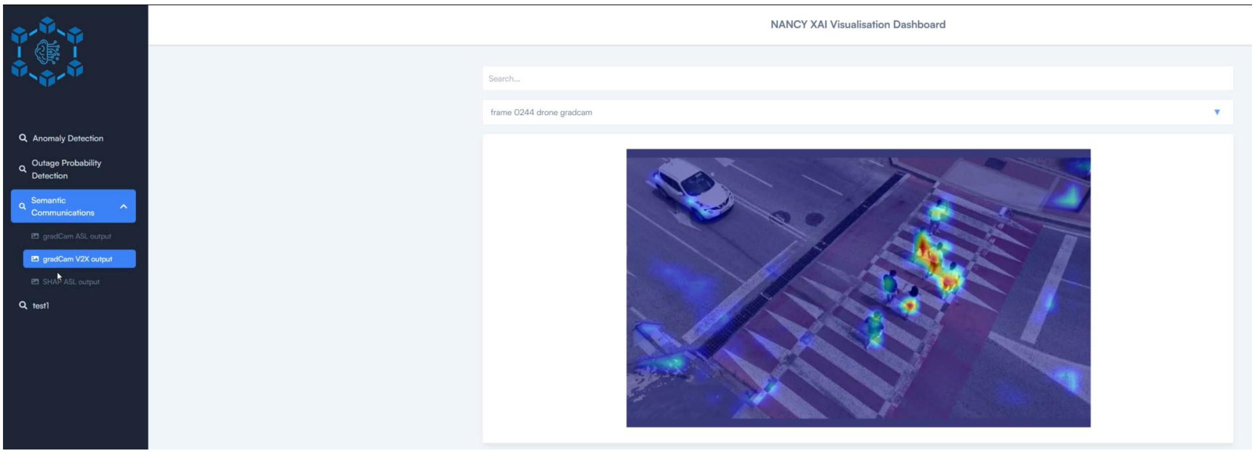


Figure A 6: Semantic communications menu – gradCam V2X output

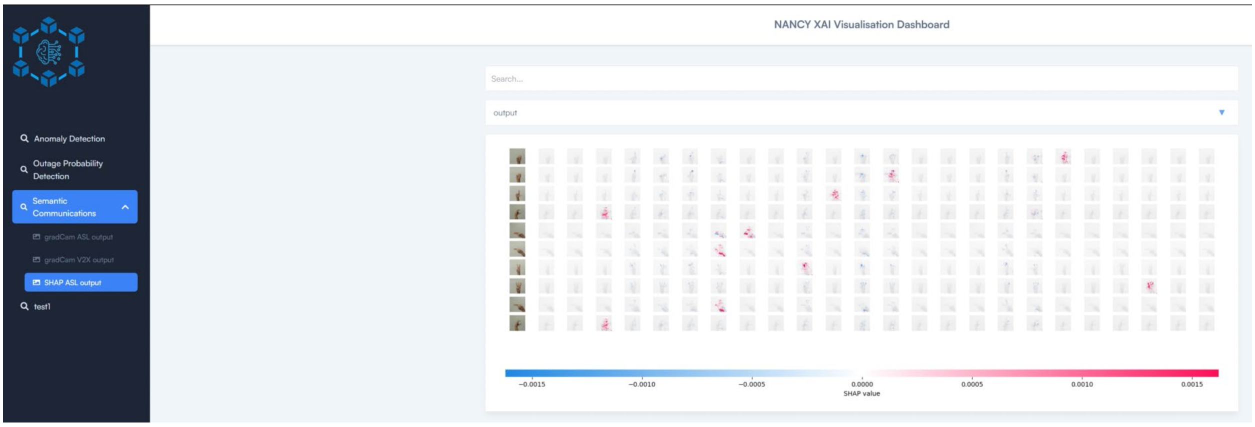


Figure A 7: Semantic communications menu – SHAP ASL output