

Smart Incident Prediction from NOC Alert Events in Digital TV Broadcasting Networks

Francisco Javier González-Serrano, Lorena Álvarez-Pérez, Harold Y. Molina-Bulla and Marcelino Lázaro

Universidad Carlos III de Madrid



Project PID2021-125652OB-I00 funded by MCIN/AEI/10.13039/501100011033 and FEDER "A way of making Europe"

Outline

1. Introduction
2. Objective
3. Alarm types
4. Smart NOC: Workflow
5. ML classifiers
6. Experimental results
7. Conclusions
8. Future lines

Project PID2021-125652OB-I00 funded by MCIN/AEI/10.13039/501100011033 and FEDER "A way of making Europe"

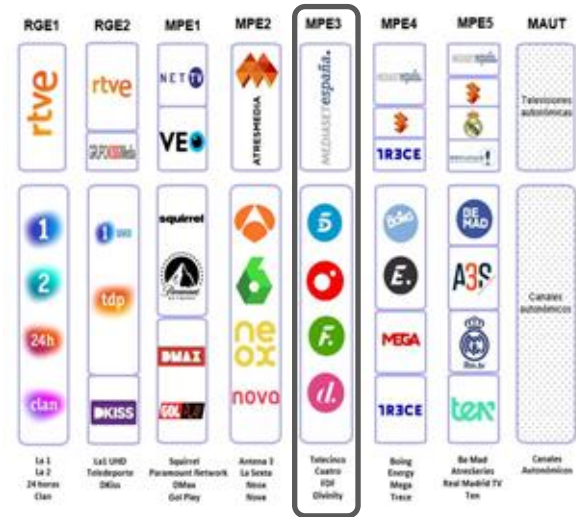


1. Introduction

Spanish Digital Terrestrial Television (DTT)

- It is organized into **multiple multiplexes**.
- Each **multiplex (Service in this work)** transmits **several TV channels** over the same frequency.
- A **failure in the Service** causes the simultaneous **outage** of all the **channels** it carries.
 - It can generate **multiple alerts or alarms**.

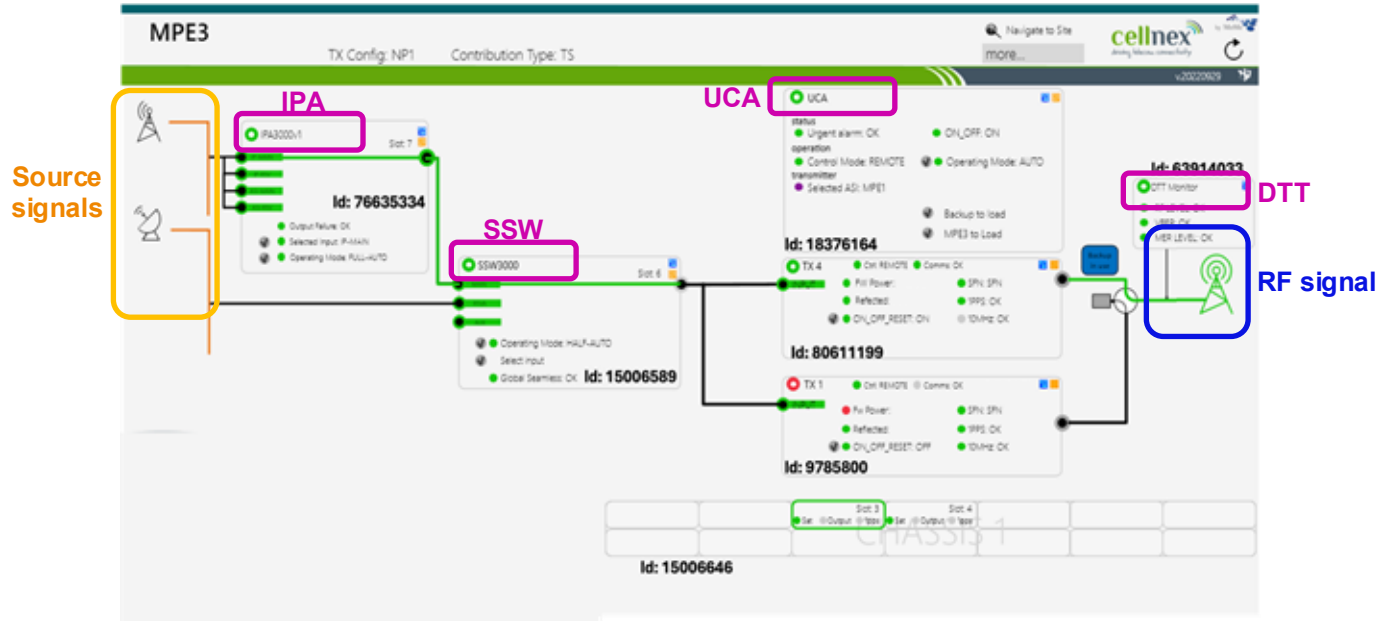
Structure of DTT channels



MPE3 Service

1. Introduction

Transmission architecture for MPE3 in Cellnex (For a particular site)



1. Introduction

- **Signal sources**

- Both **satellite and terrestrial signals** are processed by the **IPA device**, which selects the input and sends the transport stream forward.

- **Signal switching and routing**

- The signal goes through a **switching unit (SSW)**, which ensures global **seamless switching in case of failure**.

- **Monitoring and control**

- The **UCA device** operates remotely in **automatic mode**, selects the **active multiplex**, **monitors urgent alarms**, and **switches to backup if needed**.

- **Two transmission modules** (TX 1 and TX 4) are used to **provide redundancy** and **ensure continuous service**.

- **Final RF Output:**

- The signal reaches the **DTT Monitor**, which **checks quality metrics** (MER, VBER and RF level) before broadcasting.

1. Introduction

- **System overview**

- National DTT networks: Dozens of high-power transmitters and hundreds/thousands of low-power gap filters.
- Each site: satellite/terrestrial links, synchronization, RF, redundancy.

- **Current issues**

- **Frequent failures** → **High volume of alarms** and the need for efficient **NOC** operation.
- **Limitations of traditional rule-based systems:**
 - High false positive rates.
 - Operator NOC overload

1. Introduction

Network Operation Center (NOC) in Cellnex

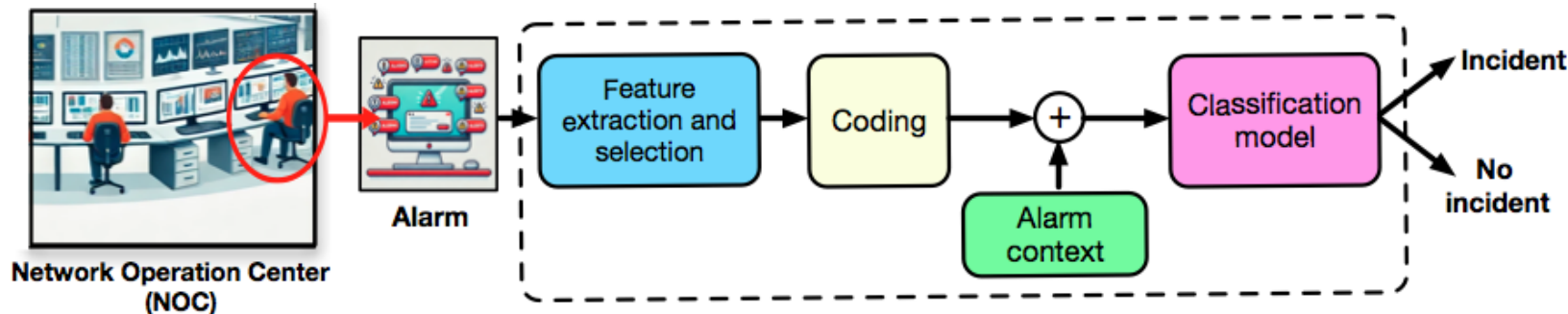


Image taken from: [smart-noc.com](https://www.smart-noc.com)

2. Objective

SMART NOC Project

- To **assist NOC operators** in detecting and resolving service interruptions.
- **Machine Learning (ML)** for automatically **predict incidents** in **DTT** using alarms and **incidents logs**.
 - **Temporal analysis** and **sequential pattern** extraction.



3. Alarm types

Element alarms

- Transmitters, encoders, and multiplexers track parameters (signal strength, bit error rate, temperature, and power).
- Measurements are sent to the Network Management System (NMS) via standard protocols (e.g., SNMP).
- Predefined thresholds → if exceeded → trigger an **element alarm**.
- Provide **early warnings** of equipment issues, enabling **proactive maintenance**.

3. Alarm types

Service alarms

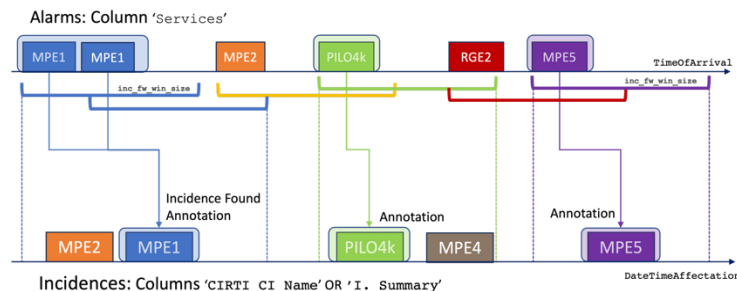
- They indicate **problems** that **directly** impact what the **viewer sees or hears**.
 - **More critical** than element alarms — they are prioritized in incident response.
- Often linked to **Service Level Agreements (SLAs)** with content providers.
- Monitor the **overall performance of broadcast services** (multiplexes), not just individual devices.
- **Triggered by operator-defined rules**, often through correlation of multiple element alarms or evaluation of service-level metrics like latency and jitter.

4. Smart NOC: Workflow (I)

- The **dataset** is based on **real alarm and incident records** from **Cellnex**.
- How to **design** this dataset?
 - **Merging**: Combine and normalize element and service alarms.
 - **Cleaning**: Standardize and complete alarm information.
 - **Content extraction**: Analyze alarm behavior at 3 levels.
 - **Site**: All services and support systems (e.g., power, climate).
 - **Service**: Alarm features for each multiplex.
 - **Equipment**: Recent alarms from individual devices.
 - **Context interval**: **Temporal** analysis and **sequential pattern** extraction (**5 or 10h-window**)
 - **Compute statistics**: count, duration, mean, std, etc.
 - Detect **frequent alarm sequences** with algorithm **PrefixSpan**.

4. Smart NOC: Workflow (II)

- **Feature coding:** Convert all data into numerical features
 - Encoding of **binary** and **categorical** features.
 - **NLP technique** for **unstructure text** (Universal Sentence Encoder).
- **Annotation:** Link alarms to incidents using time/content matching



- **Prediction:** Train **ML models** to detect (future) incidents

5. ML classifiers: Individual

Logistic regression (LR)

- **Linear model** used to **estimate** the **probability** of a sample belonging to **class positive**.
- To prevent overfitting, L2 regularization was applied.
 - Fine-tuning (cross-validation: 5-fold):
 - Regularization strength (C)
- **Bayesian MLP**
 - **Neural network** classifier tailored for **imbalanced binary classification problems**.
 - It uses a **cost-sensitive loss function** based on estimated **Bayesian decision cost**, with class priors and error costs controlling imbalance.
 - Training minimizes the cost function via gradient descent.
 - The model output is evaluated using **Parzen window estimation**.

5. ML classifiers: Ensembles (I)

- **LightGBM**

- Fast gradient boosting framework optimized for large datasets and high-dimensional features.
- It uses **GOSS** and **EFB** to reduce computation and dimensionality and employs **leaf-wise growth** with regularization to prevent overfitting.
- Fine-tuning (cross-validation: 5-fold):
 - Number of estimators
 - Number of leaves
 - Learning rate

- **RUSBoost**

- Boosting method for imbalanced data that combines random undersampling with ensemble learning.
- Each iteration trains decision tree on a different balanced subset, reducing majority-class bias and enhancing diversity.
- Fine-tuning (cross-validation: 5-fold):
 - Number of decision trees
 - Learning rate.

ML classifiers: Ensembles (II)

- **Label Switched (LS) Ensemble**

- A **fraction of majority-class samples** are **switched to minority**, and **vice versa**.
- **Each base learner** is **trained** in this **transformed label space**, improving balance and diversity in the ensemble.

The optimal decision threshold is analytically adjusted based on the switching rates α (majority $\rightarrow$ minority) and β (minority $\rightarrow$ majority), and the dataset imbalance ratio.

6. Experimental results

- **Smart NOC dataset**
 - **Real** dataset
 - Imbalance Ratio (IR): ≈ 57.3

	Contextual Info	Features	Samples			IR	Type
			C_0	C_1	Total		
Smart NOC	No context	621	43080	752	43832	57.28	Real
	Patterns	623					
	Time	729					
	Pat+Time	731					

6. Experimental results

- **Scenarios evaluated**

- **No context:** Without additional information.
- **Patterns:** It includes patterns in alarm behavior to **identify interactions** with **other elements** in the network within a **10-hour window**.
- **Time:** It contains **information related to alarms and elements** within a **10-hour window**.
- **Pat+Time:** It is a **combination of patterns and time** within a **10-hour window**.

6. Experimental results

	Ctx	Time (h)	Type		Smart NOC (IR $\approx$ 57.3)			
			Pat	Time	Acc	Bal Acc	P_{FA}	P_D
LR	No				0.9213	0.9325	0.0790	0.9441
					0.9223	0.9343	0.0780	0.9468
	Yes	10	$\times$	$\checkmark$	0.9546	0.9540	0.0453	0.9534
			$\checkmark$	$\checkmark$	0.9492	0.9513	0.0508	0.9534
MLP Bayes	No				0.9390	0.9462	0.0612	0.9537
					0.9477	0.9454	0.0522	0.9429
	Yes	10	$\times$	$\checkmark$	0.9854	0.9678	0.0140	0.9496
			$\checkmark$	$\checkmark$	0.9876	0.9670	0.0116	0.9455
LS Ensemble	No				0.9445	0.9490	0.0555	0.9536
					0.9538	0.9472	0.0459	0.9404
	Yes	10	$\times$	$\checkmark$	0.9831	0.9686	0.0163	0.9536
			$\checkmark$	$\checkmark$	0.9885	0.9714	0.0107	0.9536
LGBM	No				0.9417	0.9410	0.0582	0.9402
					0.9498	0.9300	0.0495	0.9096
	Yes	10	$\times$	$\checkmark$	0.9975	0.9713	0.0016	0.9495
			$\checkmark$	$\checkmark$	0.9974	0.9732	0.0017	0.9481
RUS Boost	No				0.9348	0.9406	0.0653	0.9467
					0.9351	0.9385	0.0650	0.9420
	Yes	10	$\times$	$\checkmark$	0.9861	0.9699	0.0133	0.9532
			$\checkmark$	$\checkmark$	0.9866	0.9702	0.0128	0.9532

Both type of contextual
information!

6. Experimental results

- **LightGBM** with **temporal context** and **pattern features** (within a **10-hour window**) achieves the best **balanced accuracy** (0.9732) and **very low false alarm rate** (≈ 0.0017).
- **Bayesian MLP**, **LS Ensemble**, and **RUSBoost** also perform well, though with slightly **higher variability** and false alarm rates.
- All models benefit from **time-based features**, improving detection rates and reducing false alarms.
- Boosting-based models + temporal context $\rightarrow$ effective for **incident prediction in imbalanced, real-world settings**.

7. Conclusions

- A complete architecture for **incident prediction in DTT networks** has been proposed, integrating alarm preprocessing and contextual analysis.
- Incorporating **temporal and pattern-based alarm context** significantly improves prediction accuracy.
- **LightGBM** achieves the best performance across all scenarios, especially in **highly imbalanced real-world data** (Smart NOC).
- **RUSBoost** also benefits from contextual features; other models (LR, Bayesian MLP) show **limited gains**.

8. Future lines

- To **validate with new real-world datasets** to assess model performance under **diverse network conditions**.
- To **include feedback from NOC operators** to verify the **correctness** of **alarm-to-incident associations**.
- To enhance the system's **robustness and long-term effectiveness** in digital TV broadcasting environments.

Smart Incident Prediction from NOC Alert Events in Digital TV Broadcasting Networks

Thank you very much for your attention!

Questions?



18th International Work-Conference on Artificial Neural Networks (IWANN'25)
A Coruña (Spain), 16th June 2025