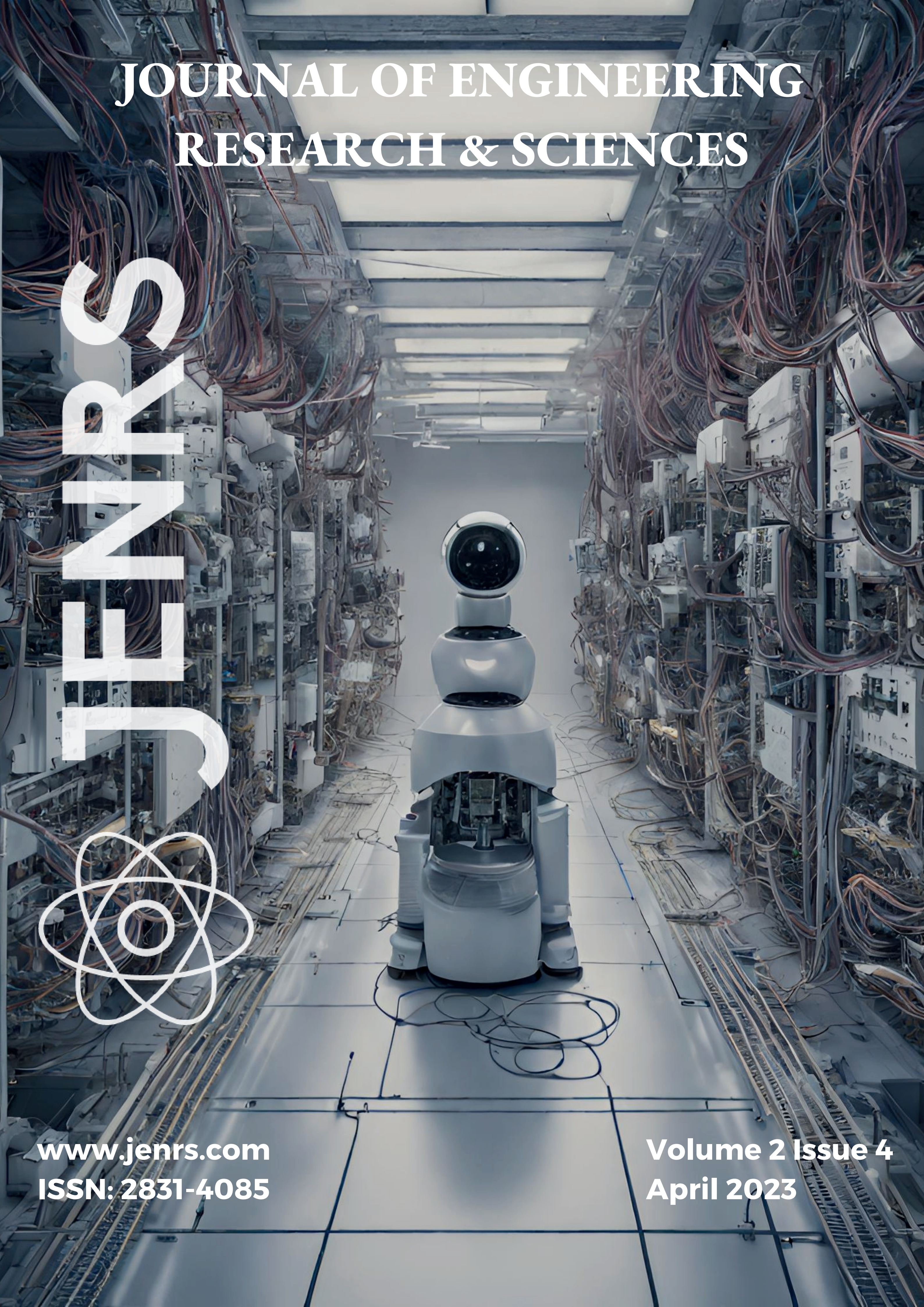
JOURNAL OF ENGINEERING RESEARCH & SCIENCES



JENRS

www.jenrs.com
ISSN: 2831-4085

Volume 2 Issue 4
April 2023

Editorial

In this edition of our journal, we feature three groundbreaking research papers that highlight advancements in network management, nonlinear circuit design, and hyperspectral image classification. Each study offers innovative methodologies and insights that address complex problems in their respective fields, contributing significantly to technological development and practical applications.

The first paper focuses on the development and evolution of a specialized tool designed to verify the health status and availability of residual bandwidth across the Lepida ScpA broadband network. This tool addresses a critical issue: ensuring that the physical bandwidth allocated corresponds to the active contractual obligations of local network operators. Previously, this verification process was manual and time-consuming. The introduction of this in-house developed tool has significantly reduced the time required for verification and provided a comprehensive overview of network status. By leveraging graph representation and well-known graph algorithms, the tool enhances the efficiency and accuracy of bandwidth verification, streamlining the process for local customers and operators [1].

The second paper introduces an advanced load-line analysis software for the design and simulation of nonlinear microwave circuits, specifically focusing on low-distortion, high-efficiency, and high-power GaN HEMT amplifiers. This software integrates DC, small-signal, and large-signal performances of GaN HEMT devices into a single package, allowing for detailed analysis of nonlinear behaviors such as AM-AM and AM-PM modulations, intermodulation distortion (IMD), and error vector measurement (EVM). Utilizing behavioral modeling and time-domain analysis, the software provides deep insights into the nonlinear characteristics of GaN HEMT devices and the design techniques for achieving low-distortion and high-efficiency amplifiers. Compared to the harmonic-balance method, this software has demonstrated comparable performance for an L-band 10W GaN HEMT amplifier, making it a valuable tool for nonlinear circuit designers [2].

The third paper explores the effectiveness of 3D Convolutional Neural Networks (CNNs) in classifying hyperspectral images (HSIs). Traditional 3D CNNs often generate an excessive number of parameters, which can hinder the extraction of spectral-spatial properties of HSIs. To address this, the study introduces a channel service module and a spatial service module to optimize feature maps and enhance classification performance. The research evaluates various CNN methodologies for HSI categorization, examining the replacement of conventional 3D CNNs with mixed feature maps to reduce spatial redundancy and expand the receptive field. The study elaborates on the efficacy of these approaches and identifies gaps in current methods, offering insights into how these gaps can be addressed to improve image classification accuracy [3].

The three papers featured in this edition exemplify the innovative and impactful research that our journal aims to publish. From optimizing network bandwidth verification to advancing nonlinear circuit design and improving hyperspectral image classification, these studies provide valuable contributions to their fields. We are honored to share these insights with our readers and anticipate that they will inspire further advancements and research.

References:

[1] G.P. Jesi, A. Odorizzi, G. Mazzin, "Graph-based Tool for Bandwidth Estimation, Health Monitoring and Update Planning in Broadband Networks," Journal of Engineering Research and Sciences, vol. 2, no. 4, pp. 1–13, 2023, doi:10.55708/js0204001.

[2] Y. Itoh, T. Kaho, K. Matsunaga, “An Advanced Load-Line Analysis Software for use in the Design and Simulation of Microwave Low-Distortion, High-Efficiency and High-Power GaN HEMT Amplifiers,” Journal of Engineering Research and Sciences, vol. 2, no. 4, pp. 14–21, 2023, doi:10.55708/js0204002.

[3] M.A. Aslam, M.T. Ali, S. Nawaz, S. Shahzadi, M.A. Fazal, “Classification of Rethinking Hyperspectral Images using 2D and 3D CNN with Channel and Spatial Attention: A Review,” Journal of Engineering Research and Sciences, vol. 2, no. 4, pp. 22–32, 2023, doi:10.55708/js0204003.

Editor-in-chief

Prof. Paul Andrew

JOURNAL OF ENGINEERING RESEARCH AND SCIENCES

Volume 2 Issue 4 **April 2023**

CONTENTS

Received: 27 December 2022, Revised: 03 March 2023, Accepted: 23 March 2023, Online: 28 April 2023

DOI: https://dx.doi.org/10.55708/js0204001

Graph-based Tool for Bandwidth Estimation, Health Monitoring and Update Planning in Broadband Networks

Gian Paolo Jesi[*,1], **Andrea Odorizzi**[1], **Gianluca Mazzini**[2]

[1]Network Department, Lepida ScpA, Bologna, 40128, Italy

[2]Engineering Department, University of Ferrara, Ferrara, 44121, Italy

[*]Corresponding author: Gian Paolo Jesi, Lepida ScpA, Via della Liberazione 15, 40128 Bologna, Italy, Email: gianpaolo.jesi@lepida.it

ABSTRACT:
This paper focuses on the genesis and evolution of our specific Company tool. It is aimed to tackle the problem of verifying the health status and availability of residual bandwidth between any node over the Lepida ScpA broadband network. In fact, there must be a correspondence between active contractual obligations signed by local network operators and the physical bandwidth which we allocate. This is the key factor that must be addressed in the early phases when processing any bandwidth requests from local customers. Before the introduction of our tool, this verification process has been carried out almost manually with a substantial cost in terms of time. The adoption of this in-house developed tool allowed us to substantially shrink of the verification time required and to provide an overview of the network status. Our tool is grounded on building a graph representation of the network and on well known graph algorithms.

KEYWORDS Graphs, Network bandwidth, Broadband

1. Introduction

The access to broadband Internet connection for citizens and companies is considered critical for the social and economical development of a modern Country. The geographical diversity of the territory of Italy created a situation in which a non negligible amount of areas suffer from poor connectivity. Unfortunately, there are cases in which these areas are not covered at all. These situations pave the road to what it is usually called as "digital divide".

Trying to limit and hopefully eliminate this problem is on top of the National and European Union (EU) agenda. At a Regional level, our company -Lepida ScpA[1]- is the main operational instrument regarding the Regional Information and Communication Technologies (ICT) Plan implementation. It has been created in 2007 by the Emilia-Romagna Regional Government (as unique shareholder and founder); currently, it has several hundreds Public Administrations (PAs) and Public Entities (PEs) as shareholders, and its activities are dedicated to them.

In order to accomplish the Plan, Lepida ScpA manages the strategies of broadband networks and several other activities such as: ensures and optimizes the delivery of ICT services and develops cloud infrastructures. In addition, it implements and manages innovative solutions for the modernization of healthcare paths in order to improve the relationship between citizens and the Regional Health Service in accordance with the provisions of EU, National and Regional Digital Agendas.

One of the core businesses of Lepida ScpA is selling its fiber optics network bandwidth at fair prices to local network operators. In turn, network operators sell an Internet connectivity service to their customers. Often, these operators *offer their service to the specific niche of customers which are located in poorly covered areas or not covered yet*.

Knowing how much bandwidth Lepida ScpA can provide from a particular network location, is just the first basic step to provide a quality service. When the customer request cannot be satisfied, it has to be aborted. In this case, it is required to plan an action in order to update the infrastructure and to satisfy similar requests in the same area ad soon as possible.

The band allocation is just one step in a wider and more sophisticated process that allows our Company to manage, update and expand the Regional broadband network.

It is important to note that bandwidth checking or monitoring here has nothing to deal with traditional real time bandwidth consumption monitoring. What really matters for us, is that when we sell some band to a customer (i.e., an operator), the sum of all bands sold must be compatible with the actual physical network capabilities of the area where the service is going to be provided.

In the last few years, the process of checking the band availability over the network had a significant evolution and lead to the creation of a specific tool having a set of continuously growing capabilities. This tool, Banda Calculus, is a building block that is going to integrate with several other tools that are on the way. In this work we are going to show the evolution of Banda Calculus and we provide the vision of our end goal in which Banda Calculus will inter operate with the other company tools which are part of the process.

Banda Calculus started as a data science notebook dealing with just one network node at a time, but now it is a standalone web application. Over the years, it become an holistic instrument capable of providing the bandwidth status of the whole network and to highlight the less capable parts or the ones already in a suffering state. The network topology is another key aspect when dealing with the healthy of a network. Identifying specific patterns that potentially lead to issues became one of the available features of Banda Calculus.

The remainder of this paper is organized as follows. In Section 2, we briefly review the current state of the art, then we discuss the specific scenario we tackle in Section 3. After presenting our algorithms and their implementations in respectively Section 4 and 5, we finally draw our conclusions in Section 6.

2. State of the Art

Since the kind of monitoring or sanity checks to the network topology are very dependent to our Company' specific needs, it is quite hard to make any comparison with existing tools.

In fact, there are plenty of monitorin] tools [2, 3] and estimation mechanisms available [4]–[5] on the market and in the open-source community which are suitable for bandwidth monitoring for example. Essentially, the idea behind these kind of tools is collecting data from network devices (such as: server hosts, routers or switches), usually via Simple Network Management Protocol (SNMP). Alerts can be set when specific parameters go beyond a predefined range. In particular, more sophisticated tools are not just limited to present charts in a dashboards, but they can also react to undesirable events by exploiting collected data with machine learning algorithms [6], [7] and making predictions.

However, our aim is different. We already have these kind of tools for monitoring network resource consumption, such as bandwidth, detecting anomalous behaviors and/or listening for alarms. Here, as stated in Section 1, *we are not interested in real-time monitoring or consumption of the bandwidth*.

Graph databases, such as Neo4J [8], are an emerging breed of tools coming from data science aimed to organize and gather data on complex structures such as graphs.

Neo4j would be ideal to build our network graph and to check its structural topology. Unfortunately, when an algorithm has to modify or add new node attributes and eventually change the structure links, it becomes highly complicated. Essentially these tools are mostly designed and optimized for querying complex structures but not for making modifications on the fly.

Since our needs are very specific and our algorithms are not just graph queries but complex procedures that shapes the structure in a specific manner, we decided to build an in-house, custom solution and to ground it on more general graph computing libraries and other high-level abstraction frameworks.

3. The General Problem

Table 1: Types of node elements in Lepida ScpA network. Unfortunately, for non-Italian speakers, many of their acronym come from their Italian name.

Acronym	Description
PAL	Lepida ScpA Access Point (Italian: "Punto di Accesso Lepida")
AG	Aggregator (Italian: "Aggregatore")
PR	Radio bridge (Italian: "Ponte Radio")
DC	Data center
MIP	Final endpoint to the core network
END POINT	Union between the DC and MIP node sets

The broadband network is make of several types of nodes (e.g., PAL, AG, PR, DCand MIP), which are listed in Table 1. This list in not exhaustive, but just the nodes significant for this paper are present.

The set of ENDPoints represent our *core* network, while the rest is the *access* part providing end-users up-links. In the core network we can manage the bandwidth by choosing between (i) tuning specific Quality of Service (QoS) strategies or (ii) upgrading the backbones. In the access network instead, where Banda Calculus comes into play, our policy is to do *not allow* any overbooking.

When an operator makes a bandwidth request, the requested band has to be booked for a specific network node, which is usually a PAL or AG node type. *The fundamental role of Banda Calculus is ensuring that the operator band requests are compatible with the current state of network bandwidth capabilities*.

The information that is adopted to build the network representation as a graph structure is mainly taken from our Network Management System (NMS). This is where the information about the whole broadband network infrastructure is stored. This knowledge is essentially maintained by human intervention through our NMS web interface. Since several people are involved in these maintenance activities, which are mostly manual, this process tends to be error prone. In this vein, our goal is to exploit Banda Calculus in order to perform sanity checks and to iron out the majority of mistakes. In fact, this focus on sanity checks is one of the latest updates we performed on our tool and we are going to address this topic in the next chapters. We exploited two (REST) APIs to ingest network data:

1. *single node oriented*: data from a single node can be queried by name[1]; it returns who are the node's immediate neighbor, details of each interface and the (total) *current bandwidth* reserved by operators. The former item is a key factor for bandwidth calculation. Unfortunately, this API has several limitations, such as: it has no access to PR nodes and it is slow.

2. *graph oriented*: it is the newest API and it has been built for the purpose of our tool. This service provides a representation of the whole network in a JSON format structure that is in turn converted into a directed graph object. Essentially, it has the same features of

[1] This is not a fully qualified DNS name, but follows an ad-hoc, internal naming scheme.

the previous API but non of its limitations. In fact, it has access to PR nodes and it is much faster, since it collects all data with a single call.

The fact of only considering a subset of network entities (see Table 1) leads to the chance of having a disconnected graph. In practice, this possibility becomes a certainty and our graph is actually disconnected and scattered in about 30 components. However, the not considered entities are not very relevant in topological terms. This allows us to have 98% of nodes inside the graph largest component, where the bandwidth algorithms are run.

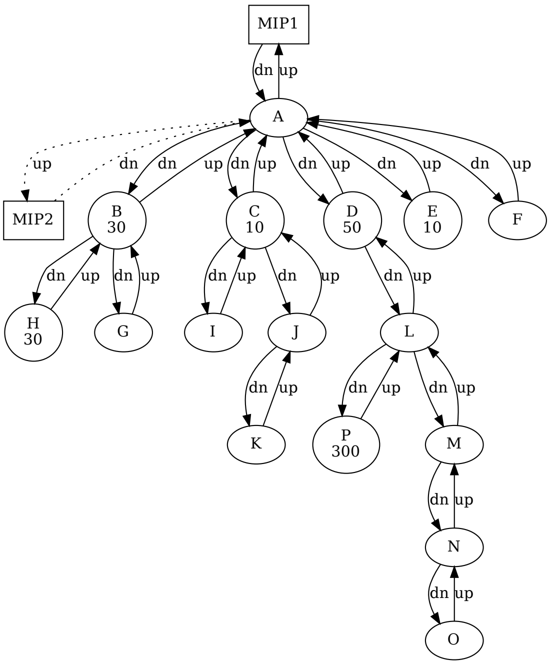


Figure 1: Network graph from which has been extracted the node L sub-graph. END POINTs are depicted as a rectangular box, while other nodes are elliptic boxes. When present, the number inside a node box represents the allocated, total operator bandwidth.

An simplified example of the particular structures (graphs) we have to deal with is provided in Figure 1, where a sub-graph for a *target node L* is shown. Suppose that an operator requests to allocate a band amount x over node L. From the whole graph, we have to extract or isolate the sub-graph in which node L is located including its neighbors and their (eventual) sub-trees in a recursive fashion. More precisely, staring from node L, we add nodes until: (a) a leaf node is found or (b) an END POINT node is found. We remind the reader that the ENDPoints set is given by the union of MIP and DC node sets (see Table 1). In order to simplify the plot even further, all links speed is set 1 Gbit/s and it is not shown explicitly.

Since we have a directed graph, each node can have inbound and outbound edges which are respectively marked as *down-link* or *up-link*. Each connection between a pair of nodes it is actually implemented by a pair of edges: an up-link and a down-link edge. The route direction of up-link edges is towards an END POINT, while the route direction of down-links is towards leaf nodes.

Every node is enriched with its current reserved operator bandwidth (op_band) if it is ≠ 0. Note that current reserved operator band parameter represents the *cumulative amount of band reserved by any operator on that node*. Dotted edges represents *inactive* links. This kind of edges usually connects a node A to one of the MIP nodes available. As depicted in Figure 1) node A is configured using an *active-standby* pattern, where the connection to MIP2 is the standby or back-up part which is exploited if and only if MIP1 link fails.

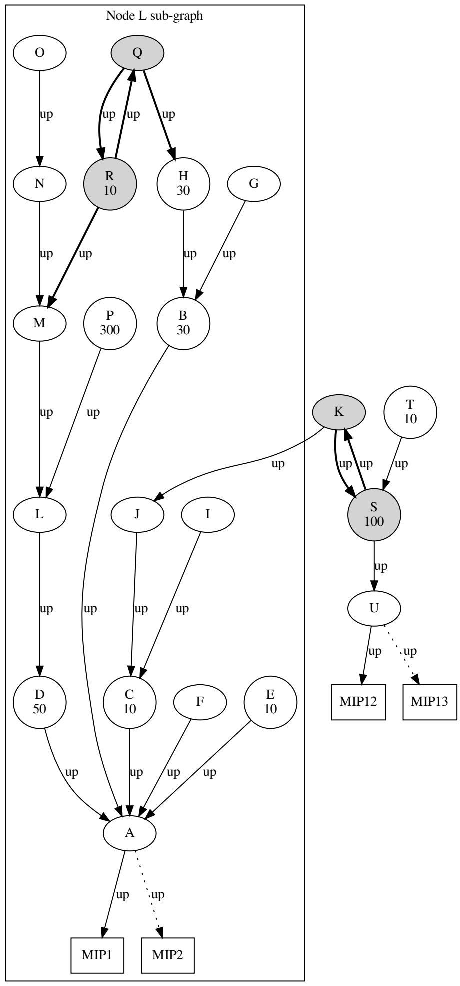


Figure 2: Prototypical, but more realistic representation of node L sub-graph. The gray filled shapes of nodes Q, R, K and S represent a *chain* structure where all edges are tagged as *up-links*. Two distinct pairs of MIP nodes are shown and are linked to distinct (PAL) nodes (i.e., A and U). The box on the left encloses node L sub-graph (LG) after the final filtering. Due to consecutive filtering processes, only up-links are left as they are exploited by the calculation algorithms.

From Figure 1 it is intuitive to understand that the available bandwidth from node L have to take into account any consumption at any node in the sub-graph; in other words, each node that stem from any down-link sub-tree might contribute to bandwidth consumption and it must be taken into account.

The general idea is to manipulate the graph structure by enriching edges with a parameter (i.e., AvailBand) which keeps track of the current band availability measured in Mbit/s.

This annotated graph is suitable to calculate the residual band between any node ad its END POINT by running any well known algorithm [9, 10]. The algorithm is going to calculate the residual band on every edge in the sub-graph (by definition) and not just over the path between a target node and its END POINT.

Unfortunately, real world conditions often present more complex scenarios. Figure 2 shows a prototypical, but realistic representation of a target node (L) sub-graph in our broadband network.

This sub-graph exhibits two main peculiarities: (i) it has two *chain structures* and (ii) two pairs of END POINT nodes. The former peculiar structure represents an exception to previous schema in which for any node pair (X, Y) we could only have one up-link and one down-link edge. Here, both edges are marked as up-links. This exception often allows the target node to reach multiple pairs of ENDPoints and this can complicate the band calculation as several (shortest) paths per END POINT becomes available. In addition, the larger the graph, the more challenging becomes its visualization and understanding.

In order to overcome these issues, we first need to clarify and impose that data traffic must follow the shortest path available to the *closest* END POINT and following the fastest links when possible. The closest MIP pair for a target node L is identified by the first steps of the algorithm. In the sub-graph, the set of nodes (LG) sharing the same closest MIP with target node L are the ones over which the actual band calculation is performed. In Figure 2, graph LG is the portion enclosed in the box.

More formally, we can express the available or residual band (RB) of node L in its sub-graph (i.e., see the box enclosed sub-graph shown in Figure 2) as:

$$RB(L) = \min\left(band\left(path_{L,ENDPOINT}\right)\right) \tag{1}$$

where the shortest path ($path_{(L,ENDPOINT)}$) is the smallest set of edges $\{e_{i,i+1}, e_{i+1,i+2}, \ldots, e_{i+(n-1),i+n}\}$ connecting L to its END POINT. The band value for each $e_{i,j}$ is the difference between the edge link physical bandwidth and the (total) operator band associated with the X_i-th node of the edge:

$$band\left(e_{i,j}\right) = phyband\left(e_{i,j}\right) - op\_band\left(X_i\right) \tag{2}$$

However, in order to address any operator band contribution from any node in the sub graph that may affect the edges over $path_{(L,ENDPOINT)}$, we must consider all shortest paths starting from any node in the sub-graph having OP_BAND $\neq 0$. Essentially, in the case depicted by Figure 2, we have to consider the following set of paths and their corresponding band contribution over each edge:

$$\begin{cases} D \overset{50}{\to} A \overset{50}{\to} MIP_1 \\ P \overset{300}{\to} L \overset{300}{\to} D \overset{300}{\to} A \overset{300}{\to} MIP_1 \\ R \overset{10}{\to} M \overset{10}{\to} L \overset{10}{\to} D \overset{10}{\to} A \overset{10}{\to} MIP_1 \\ H \overset{30}{\to} B \overset{30}{\to} A \overset{30}{\to} MIP_1 \\ C \overset{10}{\to} A \overset{10}{\to} MIP_1 \\ E \overset{10}{\to} A \overset{10}{\to} MIP_1 \end{cases}$$

By summing all instances of the same edge $e_{i,j}$ in the above schema (e.g., $D \overset{300}{\to} A + D \overset{10}{\to} A = 310$ Mbit) we obtain the total amount of band consumption over each edge. Essentially, we can rewrite (2) as:

$$band\left(e_{i,j}\right) = phyband\left(e_{i,j}\right) - \sum_{k\in I}\left(e_{i,j}\right)_k \tag{3}$$

where I represents the set of instances, as visible in the previous schema, for each individual edge $e_{i,j}$. In (3), for any edge $e_{i,j}$ we can actually calculate the (residual) band over an edge $e_{i,j}$ by subtracting all OP_BAND contributions from the physical bandwidth available on the edge link.

This approach [11, 12] allows us to calculate the available band for any target node in our broadband network no matter the complexity of the corresponding sub-graph.

3.1. *Towards an 'holistic' approach*

After being able to estimate the residual operator bandwidth for a single node in the network, we started to focus our efforts in extending the calculation to the entire broadband network. In fact, in order to monitor our network health from a topological point of view, the fact of being able to check one node at a time quickly became too limiting.

Since our basic mechanism is capable of calculation the residual band for (any) node L and since each all sub-graphs adopted for the computation are not overlapping, extending the calculus over the whole graph[2] sounds straightforward at least on paper. By adopting this approach, we can provide a global view of the bandwidth status of the broadband network and, by knowing which are the zones where band availability is *suffering* or barely sufficient, we can plan for an infrastructure upgrade.

Actually, we can move even forward.

Our tool, while searching paths over the graph structure can collect many interesting information. In particular, checking any topological issue is a natural consequence of visiting/searching over the graph. For example, we realized that the two following main issues are more common: (i) a path between two nodes is absent or (ii) a node from the sub-graph is absent. Especially the latter issue might stem from a mis-configured link property lying on the NMS which triggered a node removal from the sub-graph in one of the filtering phases.

In addition, we can build a timeline or history of the band allocation for any node and showing its evolution over time in terms of band allocation and infrastructure upgrades.

Finally, our goal is to enable the following three new features or *sanity checks* into Banda Calculus:

1. extract all *critical path*. A critical path is a path between any two nodes A and B where the available band is lower than a threshold BAND_TSD. We are interested in all critical paths according to the currently selected band threshold (BAND_TSD).

2. provide human readable information about any topological issue eventually spotted by the algorithm while visiting the graph. The fact of having *readable* information is particularly important in order to simplify the

[2]Here, we consider the largest component of the original network graph.

task of fixing the (NMS) database, since it is carried out by a group of people.

3. build a time history about operator band for each node in order to be able to keep track of any change.

4. Algorithms

Our algorithm discussion is split into three distinct parts: the first one (a) is dedicated to the residual band calculation, the second one (b) is dedicated to the holistic sanity check features, while the latter (c) is about the graph visualization algorithm.

4.1. Bandwidth Algorithm

The basic idea underlying the algorithm in order to calculate the residual band for a single node is to first annotate the graph with an AvailBand parameter and calculating the bandwidth, as previously stated in Section 3. The annotation process requires several filtering steps over the graph which are aimed to ensure data consistency and normalization. These steps are summarized as follows:

- *consistency check*: it guarantees that each record in the JSON structure coming from the NMS API contains all parameters which are relevant for the band calculation. It ensures that their values are in their corresponding ranges and are not null or NaN. A graph object G is generated at the end of this step.
- *first filtering*: from the previous polished graph G it is extracted a sub-graph SG according top a selected *target* node L. The graph SG is identified through a Breadth-First Search (BFS) over G starting from node L. The search stops when an END POINT or a leaf node is found.
- *second filtering*: SG sub-graph is refined a second time in order to just select only the relevant edges. More precisely, only edges with the following characteristics are kept:
  - edge parameter is_active is **true**
  - edge parameter template is not "NA"
  - edge parameter dir is "uplink"

  During this step, each edge is annotated with an AvailBand which is initialized to the current edge speed parameter value.
- *third filtering*: finally, graph SG is further reduced in case multiple ENDPoints are present. According to the chosen target node L, its closest MIP (or MIP pair) is selected (i.e., MIP_closest) and all nodes sharing MIP_closest as their closest END POINT are kept in SG.

  This third filtering is only applied when the actual band calculation is triggered.

The requirement to address each allocation contribution provided by nodes in any graph sub-tree as well as in chain structures, forces the algorithm to consider all shortest paths *from every node to the MIP* and not just from leaf nodes.

Figure 3 shows the calculation algorithm using a pseudo-code notation. The code does not take into account the consistency check filter. It is basically split into three parts. The first one is dedicated to the filtering processes (i.e., lines 1-9).

The second one (i.e., lines 10-14) computes all shortest paths between every node and the sub-graph ENDPoints. paths is a map or dictionary structure which collects the path set for each node. The latter part instead (i.e., lines 15-28) perform the actual graph visit and updates the AvailBand field over each visited edge. During the visit, any node having allocated bandwidth - op_band field > 0 - and being still unknown, becomes part of the already known nodes in order to guarantee an *exactly once* semantic of the algorithm.

4.2. Sanity check algorithm

The sanity checks algorithm, expressed in a pseudo-code notation, is depicted in Figure 4. The idea is simple and its actuation is scheduled at regular intervals (i.e., Δ=24 hours). For each node in the main graph (component) G we call the main function (*get_banda*()) which is the one depicted in Figure 3) which provides specific data structures required for our application needs. More precisely, it works as follows. The initialization phase (e.g, lines 1-4) prepares several data object, such as a set for basic nodes (i.e, no ENDPoints), a set for collecting topological issues and a database handle where the band history is actually stored.

The procedure runs until the node set is not empty (e.g., line 5). Nodes are pulled from the set one at a time in a (uniformly) random order and the residual band is calculated on the selection (e.g., lines 6,7). The *get_banda*() function invocation generates two structures: (a) a *banda_path* object holding the path from a target node to its MIP and the corresponding residual band and (b) a *subG* object which represents the target node sub-graph with nodes and edges enriched. Nodes are annotated with their allocated operator band, while edges are annotated with their respective residual band values.

At line 8, nodes belonging to the current target node sub-graph are removed from the node set since the band is computed for all nodes in the sub-graph. Any eventual node band update is propagated on stable storage over the database (e.g., line 9).

Finally (e.g., lines 11, 12), in case of exception, a handler manages any arising error. Three kinds of exceptions are actually trapped, which are the following:

- *NoPathException*: no path is found between node A and B, where B is a MIP. This exception may arise when there is a missing up-link edge between the two nodes. Actually, it is likely due to a mis-configuration in the NMS: in fact the edge can be present but it might have a wrong label, such as configured as 'down-link' instead of 'up-link'.
- *NoNodeFoundException*: a target node A or the MIP B is not found in the graph. The reason for this exception is likely due to the removal of this node during one

```
G ← 1st_filtering()
G ← 2nd_filtering()
nodes ← G.nodes() ⋂ ENDPoints
avail_mips ← G.sample()
known ← SET()
if avail_mips.length > 2 then
    G.filter_closest_mip(target_node)
    avail_mips ← G.sample()
end
foreach node ∈ nodes do
    foreach mip ∈ avail_mips do
        paths[node] ← G.dijkstra(node,mip,weight=SPEED)
    end
end
foreach node ∈ paths do
    used_band ← 0.0
    path ← paths[node]
    source ← node
    foreach item ∈ path do
        cur_band ← G[source][item][AVAIL_BAND]
        if G.nodes[source][OP_BAND] > 0.0 ∧ source ∉ known then
            used_band ← used_band + G.nodes[source][OP_BAND]
            known ← source
        end
        G[source][item][AVAIL_BAND] ← cur_band - used_band
        source ← item
    end
end
```

Figure 3: Residual band algorithm pseudo code.

of the filtering processes. Again it is likely due to a NMS bad configuration: in fact a node might has been removed from the graph if all its edges are marked as 'down-link'.

- *BadLinkException*: in this case the system cannot calculate any path since there are no ENDPoints available in the sub-graph. Here, it is very likely that the sub-graph MIPis connected through 'inactive' edges and this triggered its removal from the graph. Again, the underlying reason is a badly configured NMS.

4.3. Visualization Algorithm

It is important to note that this algorithm just focuses on graph visualization and it does not affect the band calculation in any manner. While the graph objects we manage are not huge, their size is in a range that poses a challenge when trying to display them into a graphic interface window. In fact, it is not unusual to deal with a node whose sub-graph is about 1000 nodes in size. This especially happens when radio bridge (i.e., PR) nodes are involved: since all their edges are marked as *uplink*, they are likely to join distinct parts (sub-graphs) of the broadband network by creating loops that are not filtered out by the standard processing that is performed.

Even a few hundreds nodes and their edges end up in chaotic plot when trying to display them. In addition, this plotting effort is quite useless because it is likely that the vast majority of the (sub) graph does not participate to the bandwidth consumption: only the set of nodes having the same closest MIP as the target node are actually involved. We remind the reader that the third filtering step is only applied when the actual band calculation takes place. Therefore, the sub-graph is not simplified yet. However, even if the sub-graph would have been simplified at this time, it might be too large as well to obtain a non chaotic plot.

In any case, when dealing with such oversize sub-graphs we need to make a decision and choosing what we are really interested to. It is not a matter to choose or design a new plotting layout. When a user have to check the residual band for a target, the goal is to see the situation over the path that links the target node to its ENDPoints. The more we move far from this path, the less is the interest and the value for the information.

The amount of available band along this path is of course dependent by the eventual contribution coming from any sub-branch at each path node, but these contributions are calculated by the previous algorithm (see: 4.1) and affect the band availability at each edge.

According to what we just stated, the basic idea underlying the visualization algorithm is to first focus on the shortest path connecting the target node to its closest MIP (i.e., or

```
repeat periodically every Δ time units                    % Code executed every Δ=24 hours
    nodes ← G.nodes() ⋂ ENDPOINTS
    G_errors ← ∅
    history ← DB.instance()
    while nodes ≠ ∅ do
        node ← nodes.sample(1)
        banda_path, subG ← get_banda(G, node)
        nodes.pop(subG.nodes())
        history.update(node, banda_path, subG)
    end
    on GraphException : ex do
        G_errors ← ⟨ ex.node, ex.info ⟩
    end
end
```

Figure 4: Algorithm pseudo code for sanity checks calculation.

```
1  K_MAX ← 50;                 % Constant: max number of nodes for visualization graph
2  avail_mips ← G.getMips()
3  mip ← avail_mips[0]
4  path ← G.dijkstra(target_node, mip, weight=SPEED)
5  simple_G ← G.subgraph(path).copy()
6  foreach item ∈ path and simple_G.size() ≤ K_MAX do
7      if not item.isMIP() then
8          extra_nodes ← G.subgraph([G.allNeighbors(item) + item]).copy()
9          simple_g ← compose(simple_g, extra_nodes, K_MAX)
10     end
11 end
```

Figure 5: BANDA CALCULUS Algorithm pseudo code for generating the simplified graph suitable for visualization.

MIP pair), then to expand this "graph-path" by adding the neighbors of each node belonging to the shortest path. Any *i*+1 level of nodes can be added by iterating the last step over the previous level of added neighbors. As a general rule, this process can stop when the graph size reaches *K_MAX* (e.g., with *K_MAX*=50).

In such a manner, we can plot what it is really required.

The actual algorithm pseudo code is depicted in Figure 5. The first four lines are in charge to detect the MIP nodes in the target node sub-graph (e.g., which is *G* in the code). Shortest paths are calculated using Dijkstra algorithm and a *simple_G* graph object is generated. This graph just contains the path items, the original sub-graph MIPs and their arcs.

The *simple_G* object is enriched by adding all neighbors for each node member of the shortest path (i.e., see lines 5-10). Any member being a MIP is skipped. The graph size is limited by the constant *K_MAX* both in the loop statement and by the commodity function *compose*() which actually merges two graphs together. More precisely, these graphs are: (i) the *simple_G* object and (ii) the graph made by the current loop node (i.e. *item*) and its neighborhood (i.e., see line 8). Any node or arc is added just once.

5. Implementation

BANDA CALCULUS is implemented in Python3 language and through the adoption of other several frameworks for specific tasks.

Actually, the evolution of BANDA CALCULUS leaded to several implementation over time since 2019 [11]. At first, it was designed as a Jupyter [13] notebook. This choice is quite common in the data science area and it turned out to be rewarding for our goal as well.

When started, this work shared many similarities with data science projects, where data sets have to be understood, analyzed and verified. For this reason, Jupyter turned out to be a stand out candidate for prototyping our tool. Usually, the best practice working with Jupyter involves trying ideas and distilling a corpus of functions or classes and use them as basic building blocks, then this process is iterated until the problem is solved.

By following this practice, we first designed the BANDA CALCULUS application as a Jupyter notebook by exploiting exploits the set of modules we distilled. By mixing code and code and formatted text elements, a notebook can simplify documentation management. The notebook implementation helps the user in how to install a Python virtual environment and the required project dependencies. In addition, the built-in documentation provides a user with a step by step explanation of (i) what he is supposed to do in order to use the tool and (ii) how it works.

Among the several solutions to manage a graph structure in Python, we selected *NetworkX*[14] library. The main reasons is that it is strongly supported by the Open Source community, it has a wide collection of state-of-the-art grade graph algorithms, and it is *symbol oriented*; in other words, any element identifier of a graph can be a symbol (e.g., a string or any complex object instead of a numeric id) and this simplifies data management. Symbolic graph libraries are not as fast as lower level ones, but the graph we are working with are in a manageable range (e.g., ~3550 nodes and ~8100 edges).

For visualizing graphs inside a notebook we adopted frameworks (e.g., *Holoviews* and *Panel*) taken from the data science world. These are high level frameworks suited for large data-sets.

Figure 6 shows the output of Banda Calculus notebook application. It performs the following logical activities:

- *Graph creation*: the application downloads a graph structure using the API and stores the corresponding graph object on stable storage labeling the file with the current date in *graphml* format. However, if the current date matches the one of any available graph file, then this file is loaded instead. When the API fails for any issue, the most recent available file is loaded.

- *Target node selection*: through a Graphic User Interface (GUI) widget the user can select the desired target node from a drop-down list containing all valid nodes found in the graph. This choice is recorded and remains set for the entire notebook.

- *Sub-graph initialization and visualization*: graph initialization and its visualization are split in two distinct notebook cells. The output of the latter is depicted in Figure 6. The first two plots (see Figure 6(a) and (b)) are respectively dedicated to the visualization of nodes and edges attributes.

  The item color code is the following: target node is yellow, ENDPoints are green and standard nodes are blue. The edge color scheme instead is given by a (linear) color gradient function based on the corresponding speed field: faster edges are towards green, while slower ones are towards red. Any part of the sub-graph can be inspected by dragging any element or zooming. Due to their average length, node names are only visualized when moving the cursor close to their shape. By exploiting this visualization the user can check the current op_band allocation and the edges speed field.

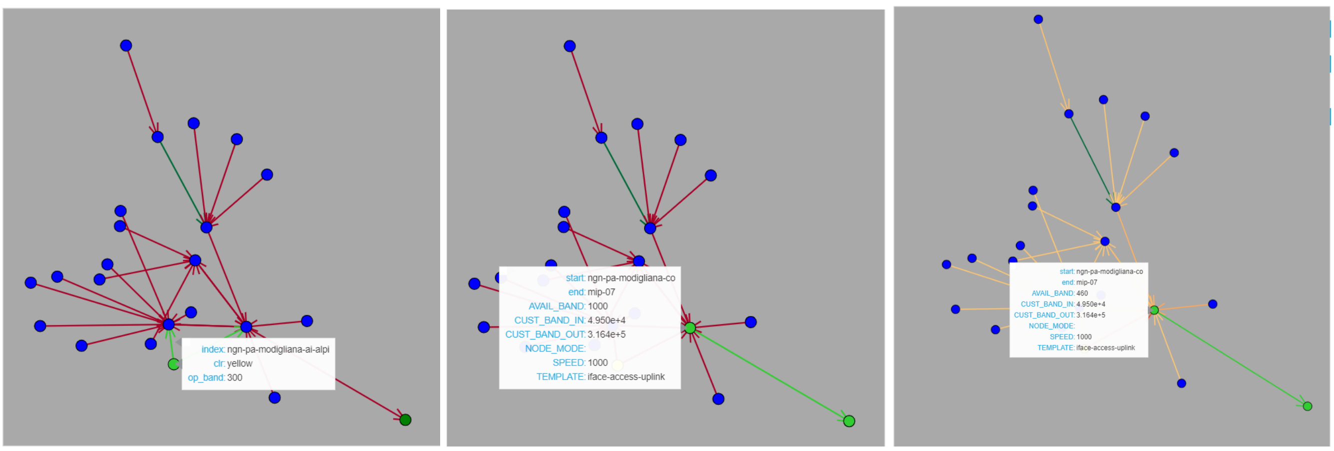
- *Operator band calculation and visualization*: the sub-plot in Figure 6(c) shows the target node sub-graph after the residual band calculation. As sub-plot (b), this visualization is edge focused. By inspecting the edge (i.e.: linking: 'ngn-pa-modigliana-co' → 'mip-07') we can see the AvailBand field annotated with the actual residual bandwidth. From the GUI it is possible to follow any path and inspecting the bandwidth at each hop. In addition, the output cell of the notebook shows bandwidth information along the path between the target node and its END POINT:

```
ngn-pa-modigliana-ai-alpi --[a. band: 700.00 Mbit/s]->
ngn-pa-modigliana-co
ngn-pa-modigliana-co --[a. band: 460.00 Mbit/s]-> mip-07
```

One of the notebook features we believe is very beneficial for our goal is the possibility to *convert it into a standalone web application* in a straightforward manner.

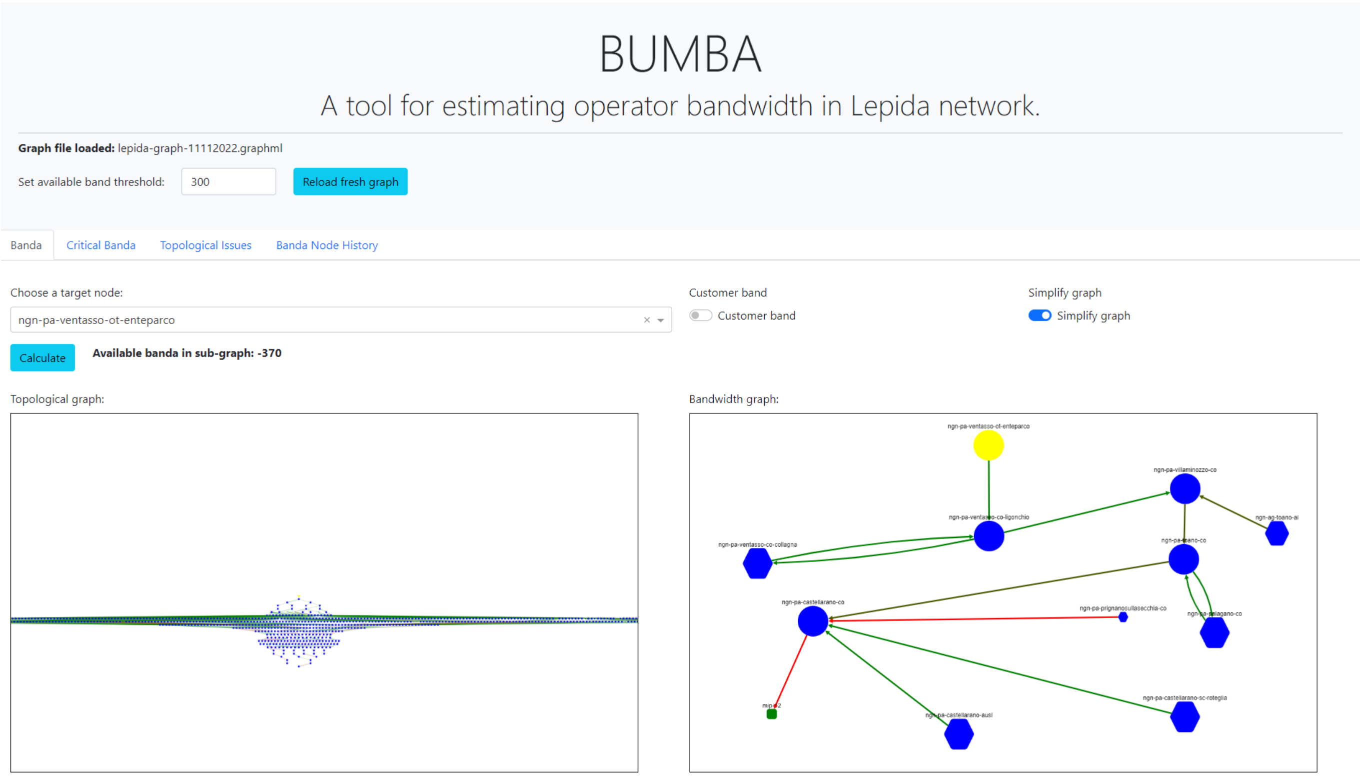
For example, this would open the road to deploy our tool in a container and serving users on the one with a traditional server approach.

Unfortunately, while the notebook app is up to the task of calculating the residual band and by exploiting the adopted frameworks it is possible to obtain a working web application with zero effort, the GUI offered by the notebook is too limiting and the documentation provided by the notebook itself is still too technical for non-tech users. Expanding and adding more sophisticated features to the notebook app is likely to became quite challenging. For this reason, we



(a) (b) (c)

Figure 6: Notebook app (sub) graph visualizations. Plots (a) and (b) are respectively dedicated to node and edges. Plot (c) instead shows the graph residual band allocation after a run of the calculation algorithm. The node color schema is the following: target node is yellow, ENDPoints are green and standard nodes are blue.



(a) (b)

Figure 7: The *Banda* tab is where the residual band is calculated for a single, selected target node. When the filtered graph is too large, as depicted in the left graph plot window (7a), by enabling the *Simplify graph* widget the graph is reduced and simplified while still retaining the information about the selected target residual band (7b).

decided to refactor our design and to switch to a different implementation capable of providing a full web application.

5.1. Web Implementation

Our implementation of the web version of the application and its 'holistic' features for network sanity checks is still grounded on Python3, but the adoption of another framework - Plotly-Dash [15] - is responsible of enabling the web interface without requiring any knowledge of standard web technologies (e.g., Javascript or CSS). Surprisingly, we managed to keep Jupyter in our design pipeline since Plotly-Dash is compatible with it and a single statement is just required to switch the application from running inside Jupyter to running standalone.

In other words, our approach of prototyping the basic functionalities into a Jupyter notebook [13] still holds.

Other frameworks has been adopted and are responsible for specific tasks such as Cytoscape [16] for rendering graphs on a web interface.

The new feature of keeping track of node band allocation over time requires some kind of database storage. Both a relational or NO-SQL approach are suitable and we decided to go for the traditional (relational) approach. In particular, we just adopted a SQL interface provided by the SQLite package and not a full database system. In fact, at the time of writing, the amount of stored data does not deserve a more sophisticated solution. However, this is likely going to change in the near future in order to improve the robustness and flexibility of the system. Since the application runs in a Docker container in our production environment, adding a database system is straightforward. Two persistent volumes has been added to the container to preserve data on stable storage which are respectively dedicated to: (a) database file and (b) daily graph files.

All sanity checks discussed in Section 3.1 are computed (once a day) when the application downloads the raw network data from the NMS API and builds an updated graph structure. Both sanity checks and the download procedure run inside a background service written in 100% Python as well. The web app exhibits the same behavior as in the previous implementation (see Section 5).

Figure 7 shows our tool GUI dedicated to the residual bandwidth calculation of a specific, single target node. In fact, note that the GUI has a first tab labeled as: "Banda" activated. The application GUI has been refreshed to better integrate the new features related to graph health.

The button labeled "Reload fresh graph" on the web Graphical User Interface (GUI) manually triggers the load of the freshest available graph from local storage. This tab performs the same task as the previous notebook application with some usability improvements.

After choosing a target node from the drop-down widget labeled "*Choose a target node*", the corresponding sub-graph is rendered in the "*Topological graph*" widget window. After applying the filtering process (until the second filter, see Section 4), the graph is still too large and its rendering provides little help to the user trying to visually verify the sub-graph topology. In these cases, enabling the switch labeled "*Simplify graph*" substantially simplifies the graph rendering and allows users to concentrate over the interesting part of the sub-graph. After enabling the switch, by pressing the *Calculate* button the bandwidth calculation is performed and this triggers to effects: (i) the amount of available bandwidth appears close to the button and (ii) the simplified graph is rendered on the *Bandwidth graph* window. By enabling the simplified rendering before choosing the target node, it would have also simplified the rendering for the *Topological graph*. Here, only the graph on the right is simplified because the switch has been enabled after the target node selection.

In graph renderings, all nodes have a rounded shape except MIP nodes, which are squared and green. Target nodes instead are yellow and any other node is blue. When graph simplification is enabled (see 'Simplify graph' switch widget in Figure 7), *border nodes* are rendered in a hexagonal shape. Border nodes are neighbors of any node being part of the shortest path starting from the selected target node. Node size is dynamic and changes according to a linear function applied to its op_band value.

The edge color meaning varies according to the particular graph window. In the topological graph widget (i.e. the left widget), the edge color represents the link speed: faster links are green, while slower links tends towards red. In the bandwidth graph widget instead (i.e., the right widget), the edge color shows the calculated residual band over that link. Its color scheme is the same as in the previous widget. Any edge connecting to a MIP node in standby mode is represented as a black dashed line and labeled by a red "standby" text on top.

Also in this application version, graph plots are dynamic and each element can be dragged and zoomed.

In addiiton to the "Banda" tab, this version has been enriched by three other tabs which respectively correspond to the graph *sanity check features*. The new tabs are named as follows: "Critical Banda", "Topological Issues" and "Banda Node History". In the following, we are going to focus on their respective interfaces.

The "Critical Banda" tab interface is shown in Figure 8. The graph plot shows the topology of a sub-graph which is the one in which lies the selected edge picked from the bottom table. The plot edges color shows their status. In fact, the selected edge is represented in red color as well as "ngn-pa-ozzano-ai-iaco" → "mip-13" edge.

The threshold that defines a critical link is towards the top of the GUI page and it stays visible no matter which tab is selected (see 7). The threshold (band_tsd) default is set to 300Mbit/s and can be overridden by editing its widget. Any edge whose residual band is less than the threshold is collected into the bottom table.

The table content can exported (in CSV format) through the "Export" button on the table top left corner.

The next tab is dedicated to topological issues and it is shown in Figure 9. Here, a table collects any exception error triggered by search algorithm over the graph (i.e., *G_errors* data structure). For each target node triggering an exception we have a table record with a corresponding "Node" column. The "Error" column contains all the required information (such as: node names, device interfaces, template description, ...) in a human readable form in order to fix the corresponding issue on the NMS. As in the previous tab, the table data can be downloaded for offline processing through the "Export" button.

Figure 10 shows the tab dedicated to node history. A user, by choosing a target node through the left drop-down widget, can query the underlying knowledge-base about any bandwidth allocation change over time. The node selection triggers the visualization of the corresponding information by populating the table on the right.

These new features allows to obtain an overview of the bandwidth status of the whole network graph and they focus on emphasizing those elements that are likely to deserve a special attention.

5.2. *Banda Calculus API Integration*

In order to support and integrate with other Company services, Banda Calculus implements a basic REST API in order to allow systems to interact together without human intervention. The API exposes a single resource via GET HTTP method. It is just sufficient to specify the target node symbol name as the only parameter. The back-end system reply is represented by a JSON structure as follows:

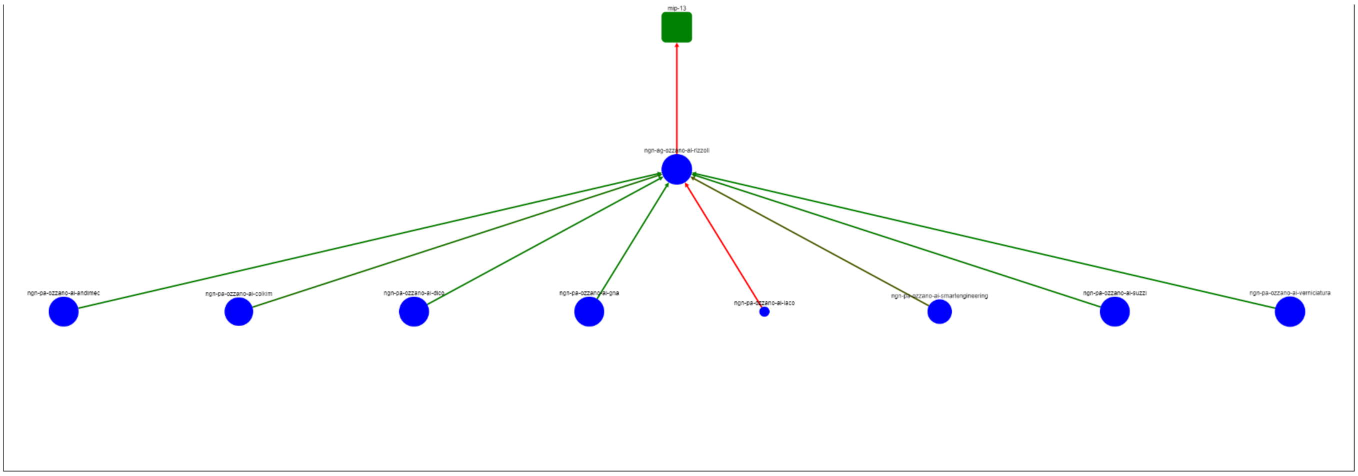
```
{
'target_node': <node_name>,
'avail_band': <band-int>,
'path_to_mip': [(<node-A>, <node-B>, <band-int>), ..., ()],
'from_graphfile': <graph_filename.graphml>,
'message': 'OK'
}
```

The object contains the target node symbol provided as parameter and the field `avail_band` shows the final result of the banda calculation process towards the closest MIP. In `path_to_MIP` field, a list of tuples depicts the exact path followed by the algorithm and specifies the residual band for each edge. Other information, such as the specific graphml file adopted as data source (i.e., `from_graphfile`) and a human readable outcome message (i.e., `message`). The former field contains the string `'OK'` or an error string in case of any issue.

6. Discussion and Conclusions

In this paper, we presented the problem we tackled when dealing with checking residual bandwidth availability in our Regional broadband network. Our ad-hoc and in house-developed solution, Banda Calculus, is coming from this particular need. Since the very beginning, the main benefit introduced by our tool is the substantial reduction (e.g., seconds versus hours) of the time required to calculate the residual available bandwidth over a specific node. Just this step turned out to be a game changer in order to provide our services to customers.

The benefits introduced by Banda Calculus are not limited to getting faster, streamlined business procedures. In fact, its evolution over time introduced several features addressing other needs. It first evolved in terms of (a) usability by becoming a web application deployed in our Intranet and in terms of (b) focusing on the whole network instead of a single node at a time. In particular, the latter avenue of evolution expanded the range of features at our disposal. These



Critical paths:

Export

Source	Avail. Band	Destination
ngn-pa-cotignola-ai-vulcanflex	-1000	ngn-ag-cotignola-ai
ngn-pa-langhirano-co	-1000	dc-pr
ngn-ag-pc	-925	mip-02
ngn-ag-pc	-925	mip-03
ngn-ag-guastalla	-730	mip-02
ngn-ag-ozzano-ai-rizzoli	-530	mip-13
ngn-pe-907	-515	mip-07

Figure 8: The *Critical Banda* tab is where paths considered *critical* from a residual band point of view are shown in a tabular format. By default, all links are ordered from most critical in a decreasing manner and the BAND_TSD is set to 300 Mbit/s as depicted in the text widget on top of the page. The selection of any table row shows triggers the sub-graph rendering in which the edge is located on the top windows for a visual examination. The table data set can also be exported in CSV format by the `Export` button.

Banda | Critical Banda | Topological Issues | Banda Node History

Error Legenda:

- **No path between X and Y** → missing uplink edge along the path between node X and MIP Y. The edge might be present, but marked as 'downlink' instead of 'uplink'.
- **Either source X or target Y is not in G** → means node not found. Bumba filtering removed node X or MIP Y from the graph. It is very likely that the filtering process has removed node X because all its in/out links are marked as 'downlink'. By removing any connection, node X is no longer part of the sub-graph.
- **A wrong INACTIVE link is likely present: no MIP DC for node: X** → the sub-graph generated by the filtering process has no ENDPOINT node (e.g., MIP). The edge with which the MIP node is connected to the rest of the sub-graph is marked as 'inactive'and this issue has removed the MIP node from the sub-graph.

Export

Node	Error
ngn-pa-piandelvoglio-ot-piazza	Either source ngn-pa-piandelvoglio-ot-piazza or target mip-07 is not in G. Misconfigured link: 'ngn-pa-piandelvoglio-ot-piazza' --[IFACE: ge0-1-0, IS_ACTIVE: True, TEMPLATE: iface-access-downlink]--> 'ngn-pa-piandelvoglio-ot-118' Misconfigured link: 'ngn-pa-montearmato-trl' --[IFACE: ge0-0-0, IS_ACTIVE: True, TEMPLATE: iface-access-downlink]--> 'ngn-pr-montearmato-sanbenedettovaldisambro'
ngn-pa-rimini-sc-lettimi	Either source ngn-pa-rimini-sc-lettimi or target mip-07 is not in G. Misconfigured link: 'ngn-pa-rimini-sc-lettimi' --[IFACE: ge0-1-0, IS_ACTIVE: True, TEMPLATE: iface-cust]--> 'ngn-pa-rimini-co' Misconfigured link: 'ngn-pa-rimini-sc-lettimi' --[IFACE: ge0-1-0, IS_ACTIVE: True, TEMPLATE: iface-cust]--> 'ngn-pa-rimini-co'
ngn-pa-lizzanoinbelvedere-trl	A wrong INACTIVE link is likely present: no MIP or DC for node: 'ngn-pa-lizzanoinbelvedere-trl'. Misconfigured link: 'ngn-pa-lizzanoinbelvedere-trl' --[IFACE: ge0-0-10, IS_ACTIVE: True, TEMPLATE: iface-access-downlink]--> 'ngn-pr-lizzano-cellmon-querciola' Misconfigured link: 'ngn-pa-casteldicasio-trl' --[IFACE: ge0-0-1, IS_ACTIVE: True, TEMPLATE: iface-access-downlink]--> 'ngn-pr-casteldicasio-porrettatermeausl' Misconfigured link: 'ngn-pa-porrettaterme-ausl-3p' --[IFACE: ge0-1-0, IS_ACTIVE: True, TEMPLATE: iface-cust]--> 'ngn-pa-porrettaterme-ausl' Misconfigured link: 'ngn-pa-lizzanoinbelvedere-trl' --[IFACE: ge0-0-10, IS_ACTIVE: True, TEMPLATE: iface-access-downlink]--> 'ngn-pr-lizzano-cellmon-querciola' Misconfigured link: 'ngn-pa-casteldicasio-trl' --

Figure 9: The *Topological Issues* tab provides users a comprehensive view of any topological issue in the current graph. The view is implemented as a table where each row shows the node which triggered some kind of error and a verbose description of the error itself. The table data set can also be exported in CSV format by the `Export` button.

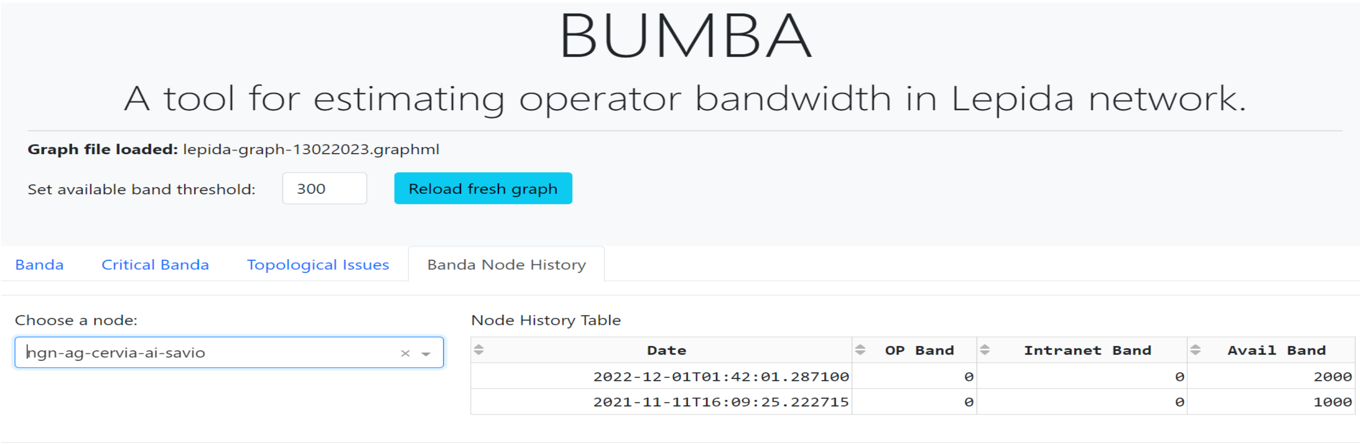


Figure 10: By first choosing a node using the combo-box widget on the left, the *Banda Node History* tab allows users to visualize the bandwidth availability for any node of the network over time.

features improve our monitoring and planning capabilities. We briefly summarize them in the following lines.

By exploiting the health monitoring features, operators can understand (c) in advance which parts of the (sub)graph might need un upgrade before it is too late (e.g., unable to provide any bandwidth). In addition, anything which is suspected of being a topological graph issue (d) is reported in a table and it is open to inspection. Also the graph visualization has been fine tuned and we adopted an in-house developed algorithm (e) that can be triggered when dealing with large graphs. Finally, in order to integrate Banda Calculus into our company processes and to allow automatic interactions between systems, we provided an API (f).

Our near future plans are actually focused on integration in order to automate and integrate as much as possible our business processes.

References

[1] "LepidaScpA Home Page", 2022.

[2] "Nagios Monitoring Solutions", 2022.

[3] "Network monitoring with intuition", 2022.

[4] V. J. Ribeiro, J. Navrátil, R. H. Riedi, R. Baraniuk, L. Cottrell, "pathchirp: Efficient available bandwidth estimation for network paths", 2003.

[5] M. Allman, "Measuring end-to-end bulk transfer capacity", "Proceedings of the 1st ACM SIGCOMM Workshop on Internet Measurement", IMW '01, p. 139–143, Association for Computing Machinery, New York, NY, USA, 2001, doi:10.1145/505202.505220.

[6] "Elastic Stack", 2022.

[7] "Graphana Labs", 2022.

[8] "NEO4J Graph Data Platform", 2022.

[9] Y. Dinitz, *Dinitz' Algorithm: The Original Version and Even's Version*, pp. 218–240, Springer Berlin Heidelberg, Berlin, Heidelberg, 2006, doi:10.1007/11685654_10.

[10] Y. Boykov, V. Kolmogorov, "An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision", "Third International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition", vol. 23, pp. 1124–1137, 2004.

[11] G. P. Jesi, G. Mazzini, "Banda calculus: a tool for bandwidth estimation in broadband network infrastructures", "2020 International Conference on Software, Telecommunications and Computer Networks (SoftCOM)", pp. 1–5, 2020, doi:10.23919/SoftCOM50211.2020.9238312.

[12] G. P. Jesi, A. Odorizzi, G. Mazzini, "Exploit company knowledge from graphs with banda calculus", "2021 International Conference on Software, Telecommunications and Computer Networks (SoftCOM)", pp. 1–6, 2021, doi:10.23919/SoftCOM52868.2021.9559101.

[13] "Project Jupyter", 2022.

[14] "NetworkX - Network Analysis in Python", 2022.

[15] "Low-Code Data Apps", 2022.

[16] "Cytoscape - Network Data Integration, Analysis, and Visualization in a Box", 2022.

Gian Paolo Jesi graduated in Computer Science in 2002 at the University of Bologna. After a short experience in an ICT Company, he rejoined his Alma Mater in late 2003 and he received his Ph.D. degree in 2007 in the area of large distributed systems.

His research work focuses on distributed/complex systems, emergent behaviors and cognition. He served as a Research Associate in several universities in Italy and Europe until he had the opportunity to join Lepida ScpA in 2018. He is author of more than 25 papers in international conferences and journals.

Andrea Odorizzi received the B.S. and MSc. degrees in Electronic Engineering (summa cum laude) from the University of Ferrara respectively in 2003 and 2005. In 2009, he received his Ph.D. from the same Alma Mater.

His research work focuses on Cryptography, Peer-to-peer multimedia applications, sensor networks. He served as PhD fellow in University of Ferrara until he had the opportunity to join Lepida ScpA in 2008. He is author of more than 20 papers in international conferences and journals.

Gianluca Mazzini graduated in Electronic Engineering (summa cum laude) and he received his Ph.D. degree in Electrical Engineering and Computer Science from the University of Bologna respectively in 1992 and 1996.

In 1996 he joined the University of Ferrara as an Assistant Professor and in 2002 he held the position of Associate Professor. His research work carried out since 1993 is related to: spread spectrum communications; applications of chaos to telecommunications; architectures for efficient radio local area networks, cellular and ambient; routing strategies in mobility sensor networks; capacity in telecommunications system; peer-to-peer networks; networks with multimedia traffic; information security. He is author or coauthor of more than 250 international publications in books, journals or conference proceedings. Google Scholar in November 2012 reports over 4700 citations with an h factor of 37 and an i10 factor of 58. His teaching shows more than 50 editions of university courses in 12 different categories. He has been the supervisor of over 140 theses and tutor for 14 Ph.D. students. He has been co-organizer of two international conferences, guest editor of the Proceedings of the IEEE, has served as Associate Editor for IEEE journals for nine years, and has served as TPC member for more than 40 international conferences. He has had roles in coordinating over a dozen projects at an international or national level, including four European projects. As first researcher in role for TLC in University of Ferrara, he founded the research group in TLC area and has established a structured series of collaborations with other organizations, including: ARCES at the University of Bologna, IEIIT at the CNR, CNIT. He has been a member of seven scientific committees and seven boards of directors or management. He was CEO of Lepida ScpA.

Received: 21 March, 2023, Revised: 05 April 2023, Accepted: 06 April, 2023, Online: 28 April, 2023

DOI: https://doi.org/10.55708/js0204002

An Advanced Load-Line Analysis Software for use in the Design and Simulation of Microwave Low-Distortion, High-Efficiency and High-Power GaN HEMT Amplifiers

Yasushi Itoh*, Takana Kaho, Koji Matsunaga

Faculty of Engineering, Shonan Institute of Technology, Fujisawa 251-8511, Japan
*Corresponding author: Yasushi Itoh, 1-1-25 Tsujido-Nishikaigan, Fujisawa, Kanagawa, Japan, itoh@elec.shonan-it.ac.jp

ABSTRACT: An advanced load-line analysis software is devised for nonlinear circuit design and simulation of microwave low-distortion, high-efficiency and high-power GaN HEMT amplifiers. A single software package can incorporate DC, small- and large-signal performances of GaN HEMT devices, and then analyze nonlinear performance of amplitude-to-amplitude (AM-AM) and amplitude-to-phase (AM-PM) modulations, and finally evaluate intermodulation distortion (IMD) and error vector measurement (EVM). High speed and high accurate simulation become available with the use of behavioral modeling for representing nonlinear performance of GaN HEMT devices. In addition, the software employs a time-domain analysis using time-varying electrical waveform and thus give clear and deep insight into the nonlinear behavior of GaN HEMT devices as well as the nonlinear circuit design technique of low-distortion and high-efficiency amplifiers. In comparison with the harmonic-balance (HB) method, comparable performances have been successfully achieved for an L-band 10W GaN HEMT amplifier.

KEYWORDS: Load-Line Analysis, Low-Distortion, High-Efficiency, Power Amplifier, Microwave, Nonlinear Circuit Analysis, GaN HEMT

1. Introduction

In recent years, low-distortion and high-efficiency of microwave high-power amplifiers represent one of the most crucial design issues in order to meet the stringent requirements of reduced cost and excellent thermal treatment of the modern wireless transmitting systems. As a starting point of power amplifier (PA) designs, the Cripps load-line theory [1] is widely used to know available output power and efficiency as well as load conditions. The Cripps load-line theory, however, has adopted the simplified device description and thus strong nonlinearity including hard saturation, large leakage current and low-frequency dispersion effects of GaN HEMT devices cannot be accurately described [2-3]. Therefore, the PA designs utilize the active and/or passive load-pull measurements as a following step to know the optimum load impedances under the actual operating conditions [4]. The load-pull measurements are, however, limited by frequency, power, impedance range, number of harmonics and stability [5]. Therefore, most of the PA designs move to the nonlinear circuit simulations using harmonic-balance method [6]. The harmonic-balance method requires the accurate nonlinear device models. Indeed, the load-pull measurement and the harmonic-balance simulation are actually a powerful tool for PA designs but only a few information on the PA designs related to low-distortion and high-efficiency can be derived. On the other hand, the load-line theory is based on time-domain waveform analysis and thus provides much useful information on load and bias conditions for low-distortion and high-efficiency.

The author has presented the nonlinear load-line analysis method to demonstrate AM-AM and AM-PM characteristics of GaAs MESFET devices in 1995 [7] and in 2001 [8]. The method, however, cannot deal with strong nonlinearity such as hard saturation and large leakage currents. Moreover, the method is based on the measured data and thus time-consuming and inaccurate simulations were crucial design issues. In order to address these design issues, behavioral modeling is utilized to represent nonlinearity of GaN HEMT devices. Moreover, the calculated AM-AM and AM-PM characteristics are represented by behavioral modeling. It makes available the 2-tone power series and envelope analyses including IP and IMD as well as EVM [9] evaluation of the modern wireless transmitting systems with high speed and high accuracy. The load-line analysis method presented here can be performed to run software written by MATLAB R2021b [10]. This is the first nonlinear load-line analysis software package ever reported. An L-band 10W GaN HEMT amplifier has been designed by using this software and compared with the harmonic-balance method [6] to make sure the validity of the software.

2. Advanced Techniques in Load-Line Analysis

2.1. Time-Varying Electrical Waveform Analysis

Principles of the load-line analysis is shown in Figure 1 [7]. Drain current $I_d(t)$ and drain voltage $V_d(t)$ swing on the load-line having a resistive slope of -gl within the area surrounded by V_k (Knee voltage), V_{br} (breakdown voltage), $V_{br}+V_p$ (V_p is a pinchoff voltage) and zero. As a magnitude of $I_d(t)$, denoted as A(J), increases with input power, the upper or the lower-half of $I_d(t)$ is clipped by I_{dss} or zero. That is, DC component of $I_d(t)$ expanded by Fourier series increases or decreases. It means that the initial bias point ***a*** (V_{do}, I_{do}) moves to a different bias point. For example, under class-AB or B operation, the lower half of $I_d(t)$ is clipped first. DC component increases and the bias point moves upward in conjunction with the load-line. Next the upper-half of $I_d(t)$ is clipped. DC component decreases and the bias point moves downward in conjunction with the load-line. This procedure is repeated until the bias point converges to some quiescent bias point ***b*** (V_{do}, I_{dav}).

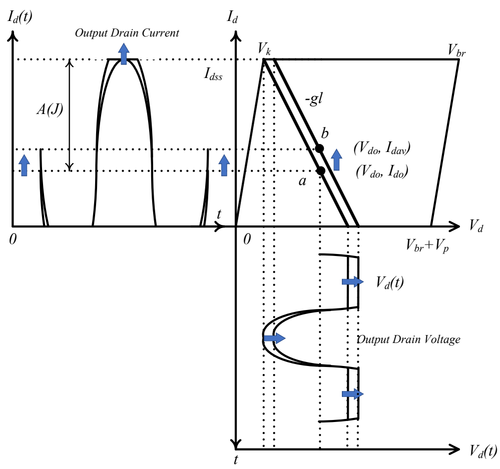
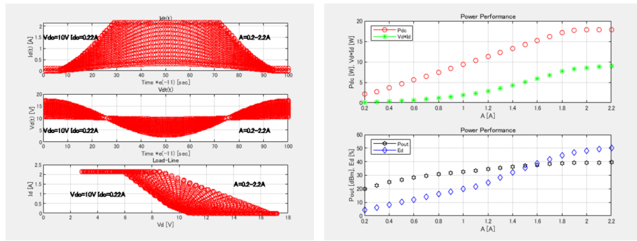


Figure 1: Principles of the load-line analysis. Drain current $I_d(t)$ and drain voltage $V_d(t)$ swing on the load-line having a resistive slope of -gl within the area surrounded by V_k (Knee voltage), V_{br} (breakdown voltage), $V_{br}+V_p$ (V_p is a pinchoff voltage) and zero. Point ***a*** is an initial bias condition (V_{do}, I_{do}). Point ***b*** is a final bias condition (V_{do}, I_{dav}).

$I_d(t)$, $V_d(t)$, a dynamic load-line are calculated by this load-line analysis software for GaN HEMT devices with V_k of 2V, V_{br} of 100V, V_p of -2V and I_{dss} of 2.14A, which is shown in Figure 2(a). As A(J) increases from 0.2 to 2.2A, $I_d(t)$ and $V_d(t)$ also increase and the bias point moves upward from the initial point (10V, 0.214A) in conjunction with the load-line. The slope of -gl can be varied as a dynamic load-line but keep constant in this case. Output power (P_{out}), drain efficiency (η_d), DC consumption power (P_{dc}) and V_d x I_d can be calculated for a variation of A(J) and plotted in Figure 2(b). As A(J) increases, P_{out} goes up to 10W and η_d also increases.



(a) (b)

Figure 2: (a) Calculated $I_d(t)$, $V_d(t)$ and dynamic load-line. (b) Calculated P_{out}, η_d, P_{dc} and V_d x I_d for GaN HEMT devices with V_k of 2V, V_{br} of 100V, V_p of -2V and I_{dss} of 2.14A

2.2. Large-Signal GaN HEMT Model Used in the Analysis

A large-signal GaN HEMT model is employed in the analysis, which is shown in Figure 3. Nonlinear circuit elements are transconductance (gm), drain-to-source resistance (R_{ds}), gate-to-source capacitance (C_{gs}) and gate-to-drain capacitance (C_{dg}), which are obtained from I-V curves as a function of the gate voltage (V_g) and the drain voltage (V_d). A forward gate current (I_{gs}) and a backward gate leakage current (I_{dg}) are also included in the analysis for hard saturation and large leakage conditions. The nonlinear circuit elements (gm, R_{ds}, C_{gs}, C_{dg}) and the gate current (I_{ds} and I_{dg}) are basically represented by behavioral modeling [11].

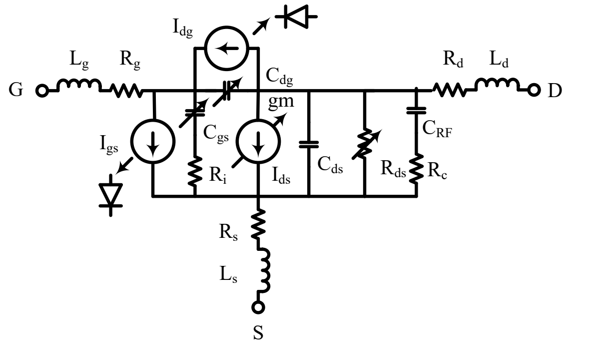


Figure 3: Large-signal GaN HEMT model. L_g, R_g, L_s, R_s, L_d and R_d are an extrinsic element, which are linear and thus keep a constant value. On the other hand, R_i, C_{ds}, C_{RF} and R_c are an intrinsic element, which are also linear and thus keep a fixed value. R_{ds} is used to represent DC characteristics. Therefore, RF characteristics are represented by $1/(1/R_{ds}+1/R_c)$ for a large value of C_{RF}.

Nonlinear circuit elements of gm, R_{ds}, C_{gs} and C_{dg} are obtained from I-V curves as a function of V_g and V_d, which is shown in Figure 4. Since $I_d(t)$ moves on the load-line with A(J), gm and g_{ds} ($=1/R_{ds}$) defined by (1) and (2) varies with A(J). Under large-signal operation, therefore, gm and gds are represented as an averaged value for one period, which are given as gm_{ave} and g_{dsave} by (3) and (4) [8]. I_d (V_g, V_d) is represented by behavioral modeling in place of the measured data for high speed and high accurate calculation. The Curtice Cubic Model [12] is used here.

$$gm = \lim_{\Delta V_g \to 0} \frac{I_d(V_g+\Delta V_g, I_d) - I_d(V_g, I_d)}{\Delta V_g} \tag{1}$$

$$g_{ds} = \lim_{\Delta V_d \to 0} \frac{I_d(V_g, V_d + \Delta V_d) - I_d(V_g, V_d)}{\Delta V_d} \tag{2}$$

$$gm_{av} = \frac{1}{T}\int_0^T gm(t)dt = \frac{1}{N}\sum_{n=1}^{N} gm(n) \tag{3}$$

$$g_{dsav} = \frac{1}{T}\int_0^T g_{ds}(t)dt = \frac{1}{N}\sum_{n=1}^{N} gds(n) \tag{4}$$

C_{gs} and C_{dg} defined by (5) and (6) also varies with A(J) on the load-line. Under large-signal operation, therefore, C_{gs} and C_{dg} are represented as an averaged value for one period, which are given as C_{gsave} and C_{dgave} by (7) and (8) [9-10]. In (5) and (6), C_{gs} and C_{ds} utilize the Statz model [13].

$$C_{gs} = C_{gs1} + \frac{C_{gs0}}{\left(1 - \frac{V_g}{V_{bi}}\right)^m} + C_{gs2}V_d \tag{5}$$

$$C_{dg} = C_{dg1} + \frac{C_{dg0}}{\left(1 - \frac{V_d - V_g}{V_{bi}}\right)^n} \tag{6}$$

$$C_{gsav} = \frac{1}{N}\sum_{l=1}^{N} C_{gs}(l) \tag{7}$$

$$C_{dgav} = \frac{1}{N}\sum_{l=1}^{N} C_{dg}(l) \tag{8}$$

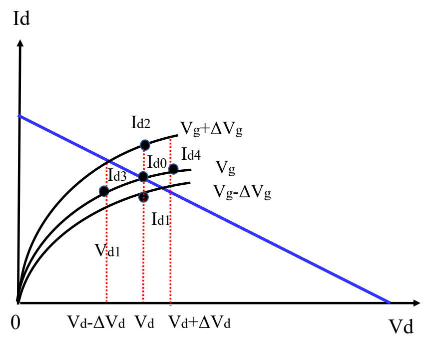


Figure 4: Nonlinear circuit elements of gm, R_{ds}, C_{gs} and C_{dg} obtained from I-V curves as a function of V_g and V_d

DC, small-signal, large-signal circuit elements as well as nonlinear capacitances consisting of Figure 3 can be read from Microsoft Excel sheet, which are shown in Tables 1(a), 1(b), 1(c) and 1(d), respectively. With the use of these data, gm_{ave}, g_{dsave} C_{gsave} and C_{dgave} are calculated and plotted in Figure 5. It is clearly shown that nonlinear elements are drastically change with A(J).

Table 1: DC, small-signal, large-signal circuit elements as well as nonlinear capacitances consisting of Figure 3

(a) 2mm_HEMT_R06.xlsx — mmHEMTR06

	A: Vk (テキスト)	B: VarName2 (数値)
1	Vk	2
2	Vbr	100
3	Idss	2.1400
4	Vdo	10
5	Ido	0.2140
6	gl	0.2675
7	N	100

(b) 2mm_HEMT_R06.xlsx — mmHEMTR06S1

	A: gm (テキスト)	B: VarName2 (数値)
1	gm	0.8963
2	cgs	1.7400e-12
3	cdg	3.0800e-13
4	cds	9.1200e-14
5	ri	0.1200
6	rds	40.8000
7	Rg	1.2300
8	Rs	1.2900
9	Rd	1.6000
10	Lg	5.0000e-12
11	Ls	3.0000e-11
12	Ld	4.0000e-12

(c) 2mm_HEMT_R06.xlsx — mmHEMTR06S2

	A: A0c (テキスト)	B: VarName2 (数値)
1	A0c	1.2000
2	A1c	0.5200
3	A2c	-0.0136
4	A3c	-0.0178
5	ac	0.7000
6	bc	0.0100
7	Vdoc	10
8	Vboc	100
9	Vbic	0.6030
10	R1c	6500
11	R2c	0
12	RF	1.7850

(d) 2mm_HEMT_R06.xlsx — mmHEMTR06S3

	A: vbi (テキスト)	B: VarName2 (数値)
1	vbi	0.6030
2	m	0.7000
3	cgs0	1.7520e-12
4	cgs1	0
5	cgs2	2.0000e-13
6	n	0.0100
7	cdg0	3.0800e-13
8	cdg1	0

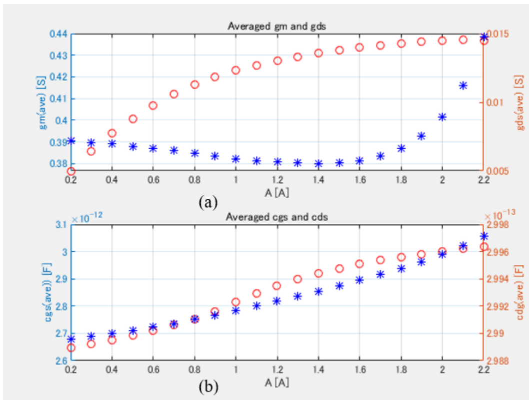


Figure 5: Calculated gm_{ave}, g_{dsave} C_{gsave} and C_{dgave}. Under class-B operation with a tuned load, I_d(t) moves partly on the load-line with zero gm. Thus, the averaged gm sometimes decreases with A(J). However, the bias point moves upward and the averaged current also increases with A(J). Then gm drastically increases.

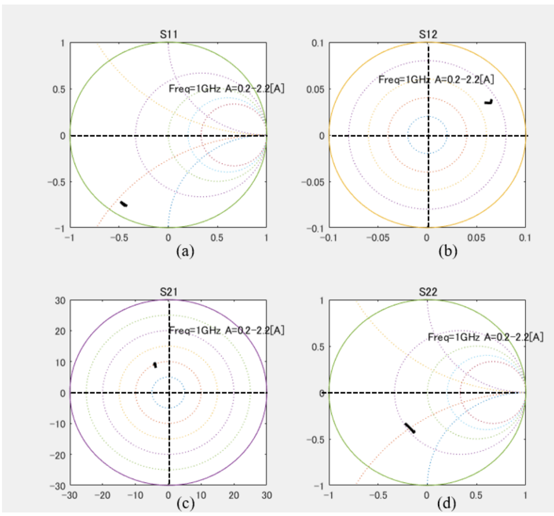


Figure 6: Large-signal S-parameters of GaN HEMT devices for A(J) from 0.2 to 2.2A at 1GHz.

Since gm_{ave}, g_{dsave}, C_{gsave} and C_{dgave} are obtained for each A(J), S-parameters of Figure 3 can be calculated, which is shown in Figure 6. A calculation was done for A(J) from 0.2 to 2.2A at 1GHz. Amid these parameters, S_{22} changes remarkably. A variation of Mag(S_{21}) and Ang(S_{21}) leads to AM-AM and AM-PM performances. In conjunction with the data in Figure 2(b), the output power (P_{out}), power gain (G_p), drain efficiency (η_d), power-added efficiency (η_{add}) and insertion phase variation ($\Delta\phi$) are calculated and plotted in Figure 7.

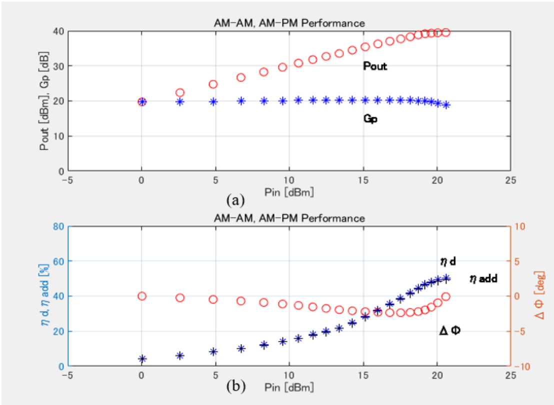


Figure 7: Calculated output power (P_{out}), power gain (G_p), drain efficiency (η_d), power-added efficiency (η_{add}) and insertion phase variation ($\Delta\phi$)

2.3. Behavioral Modeling

Once AM-AM and AM-PM performance are known, the distortion analyses including 2-tone power series analysis, 2-tone envelope analysis and EVM evaluation become available. Before the distortion analysis, AM-AM and AM-PM performances have to be represented by behavioral modeling for high speed and high accurate calculation. Behavioral modeling is listed in Table 2 [11]. The traditional distortion analysis of microwave power amplifiers deals with polynomial regression such as power series or Volterra series [1] because harmonic contents are easily handled. Thus, polynomial regression is employed here as behavioral modeling for representing AM-AM (P_{out} vs P_{in}) and AM-PM ($\Delta\phi$ vs P_{in}) performances shown in Figure 8.

Table 2: List of behavioral modeling: Behavioral modeling includes regression analysis and curve-fitting technique

Regression Analysis	Curve-Fitting Technique
Linear regression	Asymptotes slope
Logarithmic regression	Left hand technique
Power function regression	Right hand technique
Exponential regression	Build curve-fit function
Polynomial regression	Taylor's expansion
	Spline curve-fit technique

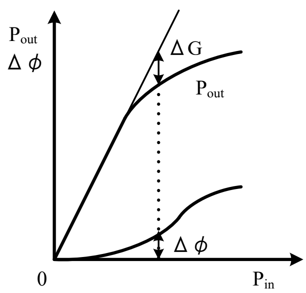


Figure 8: AM-AM and AM-PM performances of microwave power amplifiers. ΔG is a compressed gain and $\Delta\phi$ is an insertion phase variation, that is, a phase distortion

Based on the AM-AM (P_{out} vs P_{in}) and AM-PM ($\Delta\phi$ vs P_{in}) performances of Figure 7, the 3rd polynomial equations are calculated, which are shown in (9) and (10). P_{in} and P_{out} are denoted as antilog value. $\Delta\phi$ is given as degree.

$$P_{out} = a_3 P_{in}^3 + a_2 P_{in}^2 + a_1 P_{in} + a_0 \tag{9}$$
$$a_3 = -5.8643\text{E}3$$
$$a_2 = 0.5233\text{E}3$$
$$a_1 = 0.0946\text{E}3$$
$$a_0 = 0$$

$$\Delta\emptyset = a_3 P_{in}^3 + a_2 P_{in}^2 + a_1 P_{in} + a_0 \tag{10}$$
$$a_3 = -3.1400\text{E}3$$
$$a_2 = 1.2512\text{E}3$$
$$a_1 = -0.1019\text{E}3$$
$$a_0 = -0.0002$$

The calculated AM-AM and AM-PM performances shown in Figure 7 are also demonstrated in Figure 9 in conjunction with behavioral modeling. A good agreement has been achieved between the calculated and modeled data.

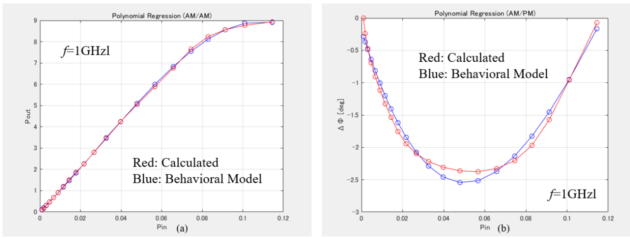


Figure 9: Calculated AM-AM and AM-PM performances combined with behavioral modeling. (a) AM-AM performance at 1GHz. (b) AM-PM performance at 1GHz. P_{out} and P_{in} are antilog number. $\Delta\phi$ is represented as degree

2.4. Distortion Analysis (2-tone Analysis)

This load-line analysis software prepares two types of 2-tone analyses: 2-tone power series analysis for weak nonlinearity and 2-tone envelope analysis for strong nonlinearity [1]. For example, in the 3rd-order 2-tone power series analysis, 2-tone signal described in (11) and (12) is inserted into (9). Then the 2nd-degree term of (9) produces the 2nd-order product at $2\omega_1$, $2\omega_2$, $\omega_1 \pm \omega_2$. The 3rd-degree term provides the 1st- and 3rd-order products at ω_1, ω_2, $3\omega_1$, $3\omega_2$, $2\omega_1-\omega_2$, $2\omega_2-\omega_1$. The 1st-, 2nd- and 3rd -order products are calculated and plotted as P_{in}-P_{out} in Figure 10. IIP_3 can be easily obtained from the intersection point of an extended linear part of ω_1 and an extended linear part of ω_3.

$$P_{in} = v_1 \cos \omega_1 t + v_2 \cos \omega_2 t \tag{11}$$

$$v_1 = v_2 = v \tag{12}$$

The 2-tone envelope analysis is shown in Figure 11 [1]. An envelope of the input 2-tone signal is modulated by a difference frequency of ω_1-ω_2 ($\omega_1 > \omega_2$). The amplified output signal is distorted in both magnitude and phase through AM-AM and AM-PM performances of PAs, which produces a serious intermodulation distortion. Input and output signals are given by (13) and (14). The input time-domain signal g(m) can be transformed from the frequency-domain signal G(k) by the inverse Fourier transformation as (15). The input signal is amplified and then the time-domain output signal g′(m) is given by (16). Finally, the frequency-domain output signal G′(k) is transformed by Fourier transformation as (17).

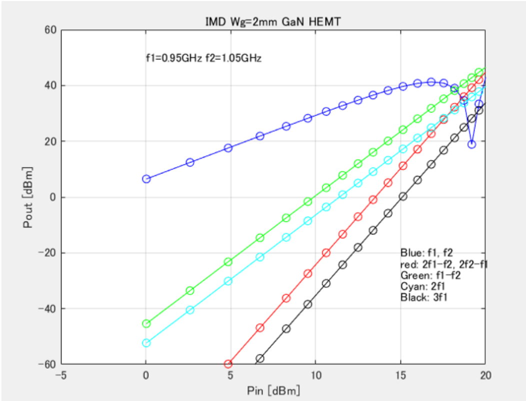


Figure 10: The 1st-, 2nd- and 3rd-order products. Red curve is the 3rd-order product ($2\omega_1$-ω_2 or $2\omega_2$-ω_1), which appear in close to carrier frequencies of ω_1 and ω_2.

$$V_i(t) = Re|\rho \cdot \exp(j\omega t)| \quad (13)$$

$$V_o(t) = Re|A(|\rho|) \cdot \exp(j\omega t + j\theta(|\rho|))| \quad (14)$$

$$g(m) = \sum_{k=0}^{N-1} G(k) exp\left(\frac{i2\pi mk}{N}\right) \quad (15)$$

$$g'(m) = |A(|g(m)|) \cdot \exp(j\theta(|g(m)|))| \quad (16)$$

$$G'(n) = \frac{1}{N}\sum_{k=0}^{N-1} g'(k) exp\left(-\frac{i2\pi nk}{N}\right) \quad (17)$$

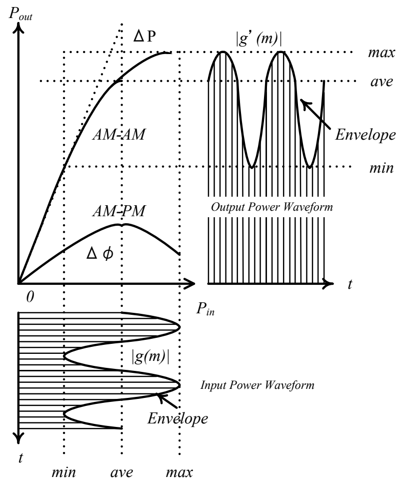


Figure 11: 2-tone envelope analysis. An envelope of input signal has a sinusoidal waveform beat by a difference frequency. An envelope of the amplified output signal is distorted in both amplitude and phase.

Time- and frequency-domain output signals are calculated with the use of 2-tone envelope analysis shown in Figure 11 and behavioral modeling of Figure 9 for 2-tone signal (f_1=0.9GHz, f_2=1.1GHz, v=0.02V) in (11) and (12), which are displayed in Figure 12. Figure 12(a) shows a time-domain output signal and Figure 12(b) displays a frequency-domain output signal (spectrum). IMD3 signals ($2f_1$- f_2 and $2f_2$- f_1) appear adjacent to carrier signals (f_1 and f_2). In addition, a difference signal (f_2- f_1), a sum signal (f_1+ f_2), 2nd-harmonic signals ($2f_1$ and $2f_2$) are also clearly shown. Due to the maximum limit of memory size of the computer, the resolution of spectrum becomes poor.

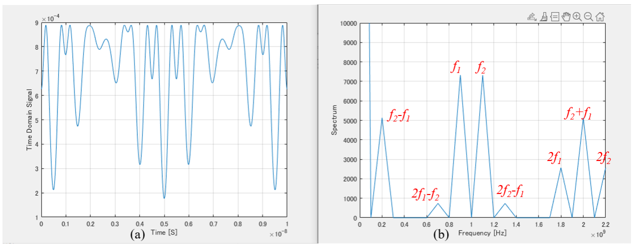


Figure 12: 2-tone envelope analysis: Time- and frequency-domain output signals are calculated for 2-tone signal (f_1=0.9GHz, f_2=1.1GHz, v=0.02V)

2.5. EVM Evaluation

Error vector magnitude (EVM) evaluation can provide a great deal of insight into the performances of digital communications transmitters and receivers [14]. The error vector is defined as a vector difference at a given time between the ideal reference signal and the measured signal, which is shown in Figure 13. AM-AM performance having ΔG and AM-PM performance having Δϕ in Figure 13(a) produce a serious vector error in Figure 13(b).

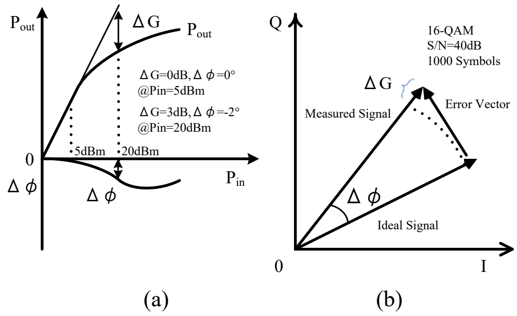


Figure 13: Error vector magnitude. (a) AM-AM and AM-PM data for use in the analysis. (b) Description on EVM schemes.

Now EVM is evaluated for GaN HEMT amplifiers having AM-AM and AM-PM performance shown in Figures 7, 9 and 13(a). EVM can be obtained by using MATLAB Simulink of EVM and MER measurement [15]. The explore model is used with an amplitude imbalance of 1 dB, a phase imbalance of 15 degrees and the DC offset of zero. Since the calculated AM-AM and AM-PM performances shown in Figures 7 and 9 cannot be used in the present form, the AM-AM and AM-PM data shown in Figure 9 are converted to a lookup table form. In the EVM analysis, 16-QAM modulated signal is used. S/N is assumed to be 40dB. EVM is evaluated at P_{in} of 5dBm for linear operation and 20dBm for nonlinear operation. Gain is 20dB at P_{in} of 5dBm and 17dB at P_{in} of 20dBm. The root-mean-square, maximum and peak values of EVM are listed in Table 3 and the constellation is demonstrated in

Figure 14. RMS value of EVM at P_{in} of 5dBm is much smaller than that of P_{in} of 20dBm, which means that a communication quality is higher because of low distortion conditions.

Table 3. Calculated root-mean-square (rmsEVM), maximum (maxEVM) and peak (pctEVM) values of EVM

Pin	5dBm	20dBm
rmsEVM[%]	2.9152	26.7234
maxEVM[%]	6.1943	40.2859
pctEVM[%]	22.8583	38.1610

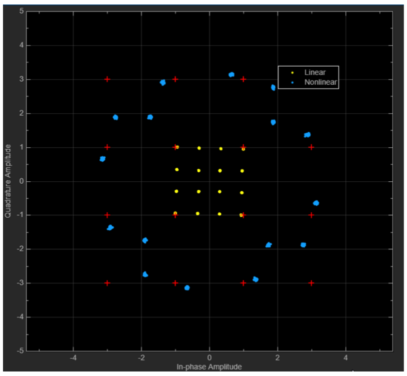


Figure 14: Constellation diagram of 16-QAM modulated signal. Red and blue dots are ideal and distorted signals at P_{in} of 20dBm. Red and yellow dots are ideal and distorted signals at P_{in} of 5dBm. It can be clearly shown that the constellation is seriously distorted at Pin of 20dBm (blue dots).

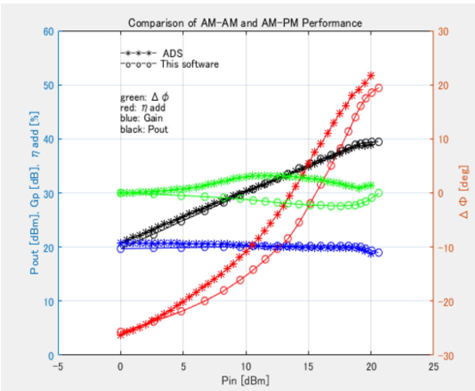


Figure 15: Comparison of the simulated power performances using the harmonic-balance simulator (ADS2021) and this load-line analysis software

3. Comparison with Harmonic Balance Method

An L-Band 10W GaN HEMT amplifier using Cree GaN HEMT CGH40010 [16] has been designed. Power performances are compared by using the harmonic-balance simulator (ADS2022 Keysight Technology) [17] and this load-line analysis software, which is shown in Figure 15. Output power and gain are in good agreement. Power-added efficiency and insertion phase variation are slightly different. These results demonstrate that the load-line analysis software introduced here is a candidate for the nonlinear analysis of GaN HEMT amplifiers. To verify the validity of the load-line software, the L-band 10W GaN HEMT amplifier is due to be actually fabricated and measured hereafter.

4. Comparative Analysis

A comparative analysis of the load-line method used in the microwave power amplifier is summarized in Table 4. A low-frequency I-V load-line measurement setup is shown in [2] and [3] to analyze low-frequency dispersion phenomena of GaN HEMT devices. The Cripps load-line theory is slightly modified to meet with low-voltage devices such as CMOS in [18] and [19]. That is, a slope of the load-line is adjusted for high efficiency in accordance with the knee voltage. By tilting a slope of the load-line for each cell of the distributed amplifier, high power and high efficiency over several octaves have been obtained [20] and [21]. The load-line is carefully chosen to achieve low-distortion and high-efficiency for both carrier and peaking amplifiers of the Doherty amplifier [22] and [23]. It must be noted that not only the load-line is carefully investigated but also time-varying waveform is checked in these load-line analyses. Similar to [2] and [3], a low-frequency I-V load-line is used to evaluate performance degradation of microwave transistor [24]. In addition, dynamic load-line is used in the design of narrowband and broadband amplifier designs [25] and [26]. This work presented here is based on a load-line analysis software, which can provide linear/nonlinear power and distortion performances. Therefore, this software can be considered to be useful to analyze various nonlinear power performance described in these References.

Table 4: Comparative analysis of the load-line method for use in the power amplifier design

Ref. No.	Year	Device	Load-line Method	Objective	Model
[2]	2009	800µm GaN HEMT	Low-frequency I-V load-line measurement	Analysis of low-frequency dispersion (i.e., traps and thermal effects)	Generic nonlinear equivalent-circuit model
[3]	2014	0.25 600 m GaN HEMT	Low-frequency I-V load-line measurement (2MHz)	Analysis of low-frequency dispersion	Behavioral Modeling of current generator
[18]	2018	GaN HEMT CGH160015D	Pedro load-line method	Impact of knee voltage effect and soft turn-on characteristic on the design of Class-B/J power amplifiers	Not described
[19]	2013	sub-micron CMOS	Extension of the load line theory to higher knee voltage value	Investigating the impact of the Knee-voltage on output-power and efficiency	Not described
[20]	2014	0.25 µm Al-GaN/GaN	Tilting load-lines	Design of uniform distributed power amplifiers having broadband high power	Not described
[21]	1987	200 µm GaAs FET	Emplying different load-line for each cell of distributed amplifier	Broadband high power distributed amplifier	Small--signal model
[22]	2008	Eudyna EGN010MK GaN HEMTs	Modulated load-line analysis	Analysis of saturated Doherty amplifier based on class-F amplifiers to maximize efficiency	OKI 0.1-W KGF1284 MESFET model
[23]	2008	Filtronic GaAs HEMT FPD750	Intrinsic load line of carrier and peak amplifier	WiMAX at 3.5 GHz is realized using a class AB amplifier	In-house Angelov non-linear model.
[24]	2021	Microwave Transistor	Low-frequency I-V and time-domain load-line measurement	Evaluation of microwave transistor degradation	Not described
[25]	2022	140nm GaN HEMT	Time-domain waveform analysis and dynamic load-line simulation	Accurate simulation of load-line and harmonic-balance methods	ASM-HEMT model
[26]	2023	GaN HEMT CGH40025F	Load-line analysis based on the series of continuum modes operation	Broadband amplifier design using class BJF-1	Not described
This Work	2023	Cree GaN HEMT CGH40010	Advanced load-line analysis for hard saturation large leakage current	Development of nonlinear load-line analysis software	Behavioral modeling

5. Conclusion

An advanced load-line analysis software for nonlinear circuit design and simulation of microwave low-distortion, high-efficiency and high-power GaN HEMT amplifiers has been presented. A single software package can incorporate DC, small-signal and large-signal performances of GaN HEMT devices, and then analyze nonlinear performance of AM-AM and AM-PM characteristics, and finally evaluate IMD and EVM. With the use of behavioral modeling, high speed and high accurate simulation become available. In addition, the software is based on a time-domain analysis using time-varying electrical waveform and thus can provide clear and deep insight into the nonlinear behavior of GaN HEMTs as well as the nonlinear circuit design of low-distortion and high-efficiency GaN HEMT amplifiers.

Acknowledgement

This research is based on a study commissioned by the National Institute of Information and Communications Technology, Japan ("Research and development on high-speed beam steering technology toward Beyond 5G" adopted number 06001).

References

[1] S. C. Cripps, *RF Power Amplifiers for Wireless Communication, 1st ED.*, Artech House, Norwood, MA, USA, 1999; pp. 24-32.

[2] A. Raffo, F. Scappaviva, G. A. Vannini, "New Approach to Microwave Power Amplifier Design Based on the Experimental Characterization of the Intrinsic Electron-Device Load Line", *IEEE Transaction on Microwave Theory and Techniques*, vol. 57, no. 7, July 2009

[3] A. Raffo, G. Bosi, V. Vadalà, G. Vannini, "Behavioral Modeling of GaN FETs: A Load-Line Approach", *IEEE Transaction on Microwave Theory and Techniques*, vol. 62, no. 1, January 2014

[4] Focus Microwaves Data Manual. Montreal, QC, Canada: Focus Microwave Inc., 1988.

[5] J. M. Cusak, S. M. Perlow, B. S. Perlman, "Automatic load contour mapping for microwave power transistors", *IEEE Trans. Microwave Theory Tech.*, vol. MTT-22, no. 12, pp. 1146–1152, December 1974.

[6] J. C. Rodringues, *Computer-aided Analysis of Nonlinear Microwave Circuits, 1st-ED.*, Artech House, Norwood, MA, USA, 1998; pp. 229-309.

[7] K. Mori, M. Nakayama, Y. Itoh, S. Murakami, Y. Nakajima, T. Takagi, Y. Mitsui, "Direct Efficiency and Power Calculation Method and Its Application to Low Voltage High Efficiency Power Amplifier", *IEICE Transaction*, vol. E78-C, no.9, pp.1229-1236, September 1995.

[8] Y. Ikeda, K. Mori, M. Nakayama, Y. Itoh, O. Ishida, T. Takagi, "An Efficient Large-Signal Modeling Method Using Load-Line Analysis and its Application to Non-Linear Characterization of FET", *IEICE Transaction*, vol. E84-C, no. 7, pp. 875-880, July 2000.

[9] Keysight Product Note 89400-14.

[10] MATLAB R2021B, Mathworks.

[11] T. R. Turlington, *Behavioral Modeling of Nonlinear RF and Microwave Devices, 1st ED.*, Artech House, Norwood, MA, USA, 2000, pp. 107-133.

[12] W. R. Curtice, M.A. Ettenberg, "Nonlinear GaAs FET Model for Use in the Design of Output Circuits for Power Amplifiers", *IEEE Transactions on Microwave Theory and Techniques*, vol. 33, no. 12, December 1985

[13] H. Statz, P. Newman, I. W. Smith, R. A. Pucel, A. Haus, "GaAs FET device and circuit simulation in SPICE", *IEEE Transactions on Electron Devices*, 1987, vol. 34, No. 2

[14] H. Jiang, P. Gong, W. Xie, B. Chen, B, H. Ma, C. Yang, "EVM Measurement and Correction for Digitally Modulated Signals", *2018 Conference on Precision Electromagnetic Measurements (CPEM 2018)*

[15] MATLAB Communication Toolbox, EVM Measurement.

[16] CGH40010, Wolfspeed, A Cree Company, USA

[17] Keysight Advanced Design System 2021

[18] X. Du, C. J. You, M. Helaoui, J. Cai, M. Ghannouchi, "Evaluation of Knee Voltage Effect and Soft Turn-on Characteristic on the Load Modulated Continuous Class-B/J Power Amplifier", *2018 IEEE MTT-S International Wireless Symposium (IWS)* 2018.

[19] P. Sandro, J. E. Mueller, R. Weigel, "Extension of the Load-Line Theory by Investigating the Impact of the Knee-Voltage on Output-Power and Efficiency", *Proceeding of the European Microwave Integrated Conferences*, pp. 488-491, 2013.

[20] M. Coers, W. Bosxh, "Analysis of Tilting Load-Line in AlGaN/GaN Broadband Uniform Distributed Amplifier", *Proceeding of GeMic*, 2014.

[21] M. L. Salib, D. E. Dawson, H. K. Hahn, "Load-Line Analysis in the Frequency Domain Distributed Amplifiers Design Examples", *IEEE MTT-S Digest*, pp. 575-578, 1987.

[22] J. Kim, J. Moon, Y. Y. Woo, S. Hong, I. Kim, J. Kim, B. Kim, "Analysis of a Fully Matched Saturated Doherty Amplifier with Excellent Efficiency", *IEEE Transaction on MTT*, vol 56, no. 2, pp. 328-338, February 2008.

[23] M. A. Medina, D. Schreurs, I. Angelov, B. Nauwelaers, "Doherty Amplifier Design for 3.5GHz Wimax Considering Load Line and Loop Stability", *Proceeding of the 3rd European Microwave Integrated Conference*, pp. 522-525, 2008.

[24] G. Bosi, V. Vadalá, R. Giofrè, A. Raffo, G. Vannini, "Evaluation of Microwave Transistor Degradation Using Low-Frequency Time-Domain Measurements", *2021 XXXIVth General Assembly and Scientific Symposium of the International Union of Radio Science (URSI GASS)*

[25] N. C. Miller, D. T. Davis, S. Khandelwal, F. Sischka, R. Gilbert, M. Elliott, R. C. Fitch, K. J. Liddy, A. J. Green, E. Werner, D. E. Walker, K. D. Chabak, "Accurate non-linear harmonic simulations at X-band using the ASM-HEMT model validated with NVNA measurements", *2022 IEEE Topical Conference on RF/Microwave Power Amplifiers for Radio and Wireless Applications (PAWR)*

[26] N. Poluri, M. M. D. Souza," Ease of Matching a Load Line Impedance in a 25 W Contiguous Mode Class BJF−1 Broadband Amplifier", *IEEE Microwave and Wireless Technology Letters*, 2023, Vol. 33, No.2.

YASUSHI ITOH received the B.S., M.S., and D.E. degrees in electronic engineering from Waseda University, Tokyo, Japan, in 1978, 1981, and 1989, respectively. He joined the Tokimec Inc. in 1981, where he has worked on the research and development of broadband low-noise amplifiers and oscillators. In 1990, he joined the Mitsubishi Electric Corporation, Information Technology R&D Center, where he has been engaged in the research and development of microwave and millimeter-wave low-noise and high power MMIC amplifiers and solid-state power amplifiers. In 2003, he joined the Shonan Institute of Technology, Kanagawa, Japan as a professor of the Faculty Engineering. From 2000 to 2003 along with 2020 to 2022, he has been a visiting professor of the University of Electro-Communications, Tokyo, Japan. From 2016 to 2018, he has been also a visiting professor of the National Chiao Tung University, Hsinchu, Taiwan. In 2000, he received the Electronics Society Award from

IEICE Japan. He is s Senior Life Member of the IEEE. He is currently a researcher of the Shonan Institute of Technology.

TAKANA KAHO is a Professor of Department of Electrical and Electronic Engineering, Shonan Institute of Technology. She received the B.S. and M.S. degrees in physics from Tokyo Metropolitan University, Japan, in 1994 and 1996 respectively. She received the Dr. Eng. degree in communication engineering from Tokyo Institute of Technology, Japan, in 2007. From 1996 to 2019, she was engaged in research on satellite equipment and MMICs at NTT Laboratories. Since 2009, she has been an Expert Committee Member on the Information and Communications Council of the Ministry of Internal Affairs and Communications, Japan. From 2010 to 2012, she was a Visiting Associate Professor at Research Institute of Electrical Communication, Tohoku University, Sendai, Japan. From 2014 to 2017, she has been a Visiting Associate Professor at Graduate School and Faculty of Information Science and Electrical Engineering, Kyushu University, Fukuoka, Japan. From 2016 to 2019, she was a Visiting Professor at Graduate School of Information Science and Technology, Osaka University, Osaka, Japan. Dr. Kaho is a senior member of the IEICE. She received the Japan Microwave Prize at the 1998 Asia Pacific Microwave Conference and the Young Researchers' Award in 2004 presented by IEICE. She received a Best Paper Award from IEICE in 2015, and 2017.

KOJI MATSUNAGA received the B.S. and M.S. degrees in physics from University of Tsukuba, Ibaraki, Japan, in 1987 and 1989, respectively, and the Dr. Eng. degree in electronics engineering from Kyoto University, Kyoto, Japan, in 2009. He joined NEC Corporation, Kanagawa, Japan, in 1989, where he was engaged in the research and development of III–V compound high frequency semiconductor devices. Since 1992, he has been engaged in the research and development of microwave and millimeter-wave high power amplifiers, wireless systems using GaAs or GaN III–V compound high frequency semiconductor devices. From 2013 to 2018, he also advanced research in power electronics. Since 2021, he has been with the Department of Electrical and Electronics Engineering, Shonan Institute of Technology, Kanagawa, Japan, where he is currently a professor. He received the Ohm Technology Award in 2011. His current research interests include microwave integrated circuit designs and wireless power transmission systems. Dr. Matsunaga is a member of the IEEE Microwave Theory and Techniques Society. He is also a senior member of the Institute of Electronics, Information and Communication Engineers, Japan.

Received: 09 September, 2022, Revised: 03 February 2023, Accepted: 31 March, 2023, Online: 28 April, 2023

DOI: https://doi.org/10.55708/js0204003

Classification of Rethinking Hyperspectral Images using 2D and 3D CNN with Channel and Spatial Attention: A Review

Muhammad Ahsan Aslam[*,1], **Muhammad Tariq Ali** [2], **Sunwan Nawaz** [1], **Saima Shahzadi** [3], **Muhammad Ali Fazal** [2]

[1] Institute of Computer Science, Khwaja Fareed University of Engineering & Information Technology, RYK, 64200, Pakistan

[2] IT Department, Khwaja Fareed University of Engineering & Information Technology, RYK, 64200, Pakistan

[3] Computer Science Department, University of Agriculture, Faisalabad, 38000, Pakistan

* Corresponding author: Muhammad Ahsan Aslam, +923206280582, ahsan.aslam4415@gmail.com

ABSTRACT: It has been demonstrated that 3D Convolutional Neural Networks (CNN) are an effective technique for classifying hyperspectral images (HSI). Conventional 3D CNNs produce too many parameters to extract the spectral-spatial properties of HSIs. A channel service module and a spatial service module are utilized to optimize characteristic maps and enhance sorting performance in order to further study discriminating characteristics. In this article, evaluate CNN's methods for hyperspectral image categorization (HSI). Examined the replacement of traditional 3D CNN with mixed feature maps by frequency to lessen spatial redundancy and expand the receptive field. Evaluates several CNN stories that use image classification algorithms, elaborating on the efficacy of these approaches or any remaining holes in methods. How do improve those gaps for better image classification?

KEYWORDS: Hyperspectral, Image classification, Deep learning, Convolutional neural network, Feature extraction, Spectral-spatial features, Machine Learning

1. Introduction

Due to the rapid advancement of optics and photonics, hyperspectral sensor nodes have been placed on numerous spacecraft. Pollution prevention, disaster prevention and control, and mineral deposit identification [1-3] are just a few of the fields where HSI categorization has gotten a lot of attention. HSI classification jobs, however, face numerous obstacles due to the huge number of spectral bands. In addition to significantly improved data and high computational cost, the Hughes phenomenon is the most remarkable challenge. One of the most effective solutions to these issues is feature extraction. However, problems like spectral variability [4] make the feature extraction operation extremely difficult. The challenge of labelling each pixel in a hyperspectral image is a vital but difficult undertaking. It allows to distinguish between distinct things of interest in a picture using the rich spatial–spectral information contained in hyperspectral photographs. Precision agriculture, environmental monitoring, and astronomy are just a few of the sectors where they've been extensively used [5]. For example, they suggested a linear mixture model for determining the mineralogy of Mars' surface by integrating multiple absorption band approaches on CRISM.

A growing body of research is being done on the categorization of hyperspectral images. Because they account for the broad spectrum of information [6] acquired in hyperspectral images [7] and reduce the dimensionality of hyperspectral images using the Locality Adaptive Discriminant Analysis (LADA) algorithm, traditional image classification methods like support vector machine (SVM) [7], [8] and K-nearest neighbour (KNN) classifier have achieved respectable performance for this task. There are other additional methods for addressing this issue. For instance, [9] offered a dimensionality reduction approach for classification of hyperspectral images using the manifold ranking algorithm as the band selection method. Additionally, they created a special dual clustering-based band selection method for classifying hyperspectral images. Although it has been demonstrated that these techniques are more successful at classification, they are unable to categories hyperspectral pictures in complicated situations.

Convolutional neural networks (CNNs) [10–12]-based algorithms have lately exhibited extraordinary performance for various tasks related to image analysis, such as picture categorization and object identification, thanks to the enormous success of deep learning. When categorizing hyperspectral images, it is important to take both the spectral and spatial perspectives into account. A hyperspectral picture, also known as the spectral perspective, is conceptually made up of hundreds of "images," each of which represents a very narrow wavelength band of the electromagnetic spectrum (visible or invisible). The 2-dimensional spatial data in the hyperspectral images of the objects, on the other hand, is covered by the spatial perspective. As a result, hyperspectral pictures are frequently represented using 3D spectral-spatial data.

1.1 Convolutional Neural Network

The ability of conventional machine-learning algorithms to assess natural data in its raw state has been constrained. It took years of careful planning and extensive domain knowledge to create a classifier that transformed raw data (such as image pixel values) into an appropriate internal representation or extracted features from which the learning new module, frequently a classifier, could identify or classify patterns in the input. Deep-learning methods are demonstrative techniques that shift recognition from a lower, more fundamental level (beginning with the raw input) to a higher, more complex one using straightforward but non-linear modules. By integrating enough of these adjustments, very complicated functions may be learnt. Higher layers of representation in classification tasks highlight characteristics of the input that are crucial for differentiating while suppressing inconsequential variations. A picture is composed of a matrix of image pixels, and the first layer of representation's learned characteristics are generally the presence or absence of boundaries in the image at specified orientations and locations. The second layer detects motifs by looking for certain patterns in data edges, independent of slight edge location discrepancies. The third layer may aggregate motifs into larger groupings that correlate to components of detection and measurement, with subsequent layers identifying items as a mixture of these pieces. Because multiple layers of features are acquired from information using a broad learning process rather than being created by people, deep learning is differentiated from other types of learning [13]. Convolutional neural networks have made achievements in a variety of pattern recognition fields during the previous decade, from image analysis to speech recognition. CNNs have the largest benefit in that they decrease the number of parameters in an ANN. This success has inspired researchers and doctors to consider larger models to address challenging issues that were previously unsolvable with conventional ANNs.

The fundamental presumption about the issues that CNN addresses is that they shouldn't have spatially dependent aspects. To put it another way, don't have to worry about where the faces are in the photographs in a facial recognition program. It doesn't matter where they are in the surroundings; their discovery is the only thing that matters. Another crucial property of CNN is its ability to extract abstract properties when fed into advanced stages or deeper levels. For instance, in the first layer of picture classification, the edge may be detected, then simpler forms in the second layer, and finally higher-level characteristics [14]. Figure 1 provides an explanation of convolutional neural networks. A popular form of neural network is the CNN [15]. A CNN is similar to a multilayer perceptron (MLP) in concept. The activation function of every neuron in the MLP labeled with input and output weights. When add extra hidden layers after 1st layer to MLP, then it is called deep MLP. Similarly, CNN is regarded as an MLP with a unique structure. The architecture of the model permits CNN to be both translation and rotation invariant because of this particular structure [16]. In a CNN design, a convolutional layer, a pooling layer, and a comprehensively layer with a corrected activation function [17] are the three essential layers.

There are other methods for hyperspectral image classification that are competing in the literature. Some of these include:

1. Support vector machine (SVM)
2. Random forest (RF)
3. Principal Component Analysis (PCA)
4. Independent Component Analysis (ICA)
5. Deep Belief Networks (DBN)
6. Convolutional Auto encoder (CAE)
7. Generative Adversarial Networks (GANs)

The reason why Convolutional Neural Networks (CNNs), 2D and 3D CNNs, and hyperspectral imaging are encouraged in the literature is due to their ability to effectively capture the spectral and spatial information in hyperspectral images, leading to improved classification accuracy. In a variety of computer vision applications, such as picture classification, object recognition, and semantic segmentation, CNNs have demonstrated exceptional performance., among others. Additionally, 2D and 3D CNNs have been designed to take into account the spatial and spectral dimensions of hyperspectral images, leading to improved performance in hyperspectral image classification tasks. In comparison, traditional methods such as SVMs, RF, PCA, ICA, DBN, CAE, and GANs may not be as effective in capturing the complex relationships

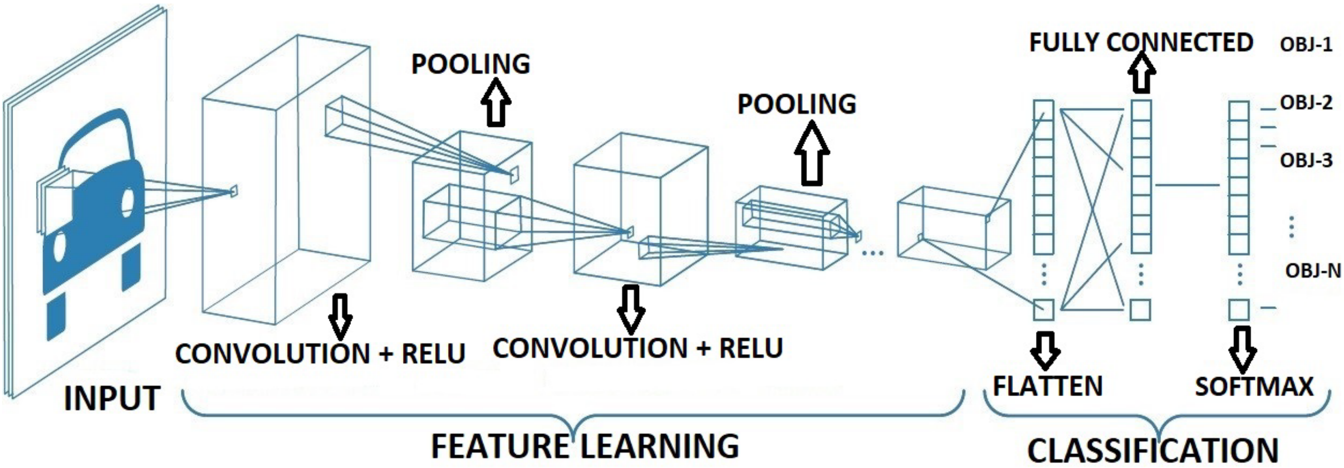


Figure 1: Understanding Convolutional Neural Network (CNN)

between the spectral and spatial information in hyperspectral images, leading to lower classification accuracy. However, these methods still have their own advantages and are often used in combination with CNNs to address specific limitations and improve performance in hyperspectral image classification tasks.

CNN is a deep learning architecture that uses layers to classify things. It also included layers labelled as one input layer, numerous hidden layers, and one output layer. CNN works in the same way that DNN does, in that it takes input from a dataset, applies functions to it in the hidden layers, and then finds the result and displays it in the output layer. Max pooling, convolution, and fully linked layers are the most commonly employed CNN layers. The filter is convolved with input information in the layer of convolution.

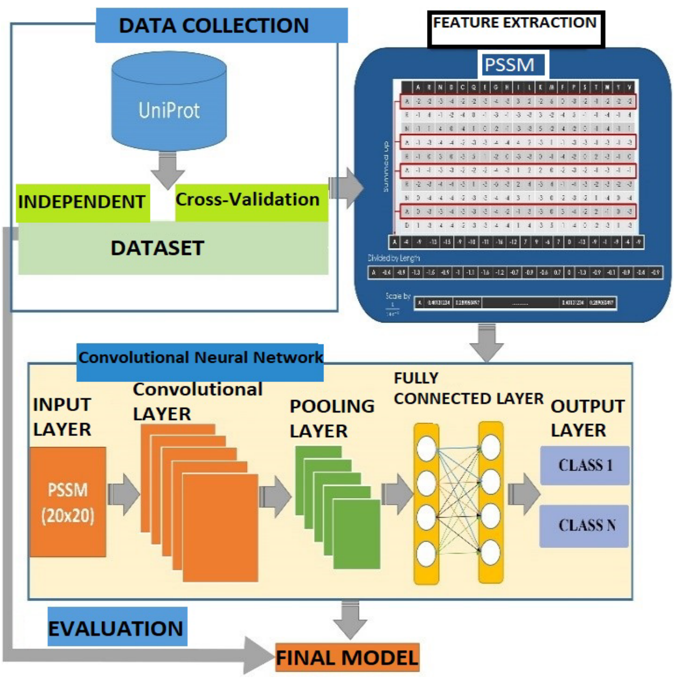


Figure 2: A 2D CNN Architecture for proposed Dataset to classify the images into different classes classification.

The input is down sampled by the max pooling layer, and the input is fully connected by the fully connected layer, which connects all neurons from the previous layer to each other [18]. Now discuss the models of CNN that are in the form of 2D, 3D, and many more, but the focus will be on some major ones that are used for some classifications. By raising the number of layers, CNN is able to learn high-level hierarchical features. When the number of layers is increased, however, the input data or gradient starts to disappear. A more value-showing model, known as a dense convolutional network, was developed to overcome this problem (DenseNet). They devised a feed-forward algorithm that can interconnect each layer with every layer. Now expanded the feature set after being inspired by the thought that dense connections can boost feature utilization. For gesture recognition, 2-dimensional DenseNet to 3-dimensional DenseNet is used [19].

The spatial perspective, on the other hand, refers to the 2D spatial data about the objects that is present in hyperspectral images. As a result, hyperspectral pictures are frequently represented using 3D spectral-spatial data. As a result, the literature has provided a variety of approaches. Contrarily, current CNN-based algorithms [20] that only pay attention to spectral or spatial information are forced to ignore the connections between the spatial and spectral viewpoints of objects captured in hyperspectral pictures [21, 22].

To extract features from these planes using three 2D CNNs, and then integrate three 2D network architectures in parallel, resulting in the multichannel 2D CNN. The 2D CNN model is made up of three elements of the 2D CNN architecture running in parallel, as well as a fully connected hidden unit that integrates multichannel data. Each 2D CNN takes only one sort of multichannel 2D image as an input and performs convolution computing on its own. The outputs from three 2D CNN sections are flattened, concatenated, and then fed into a fully connected neural network for learning. Finally, 2D CNN produces the categorization outcome. Given that the concatenation characteristics include features obtained from three orthogonal planes, 2D CNN considers 3D. As above figure 2 shows the architecture of 2D convolutional

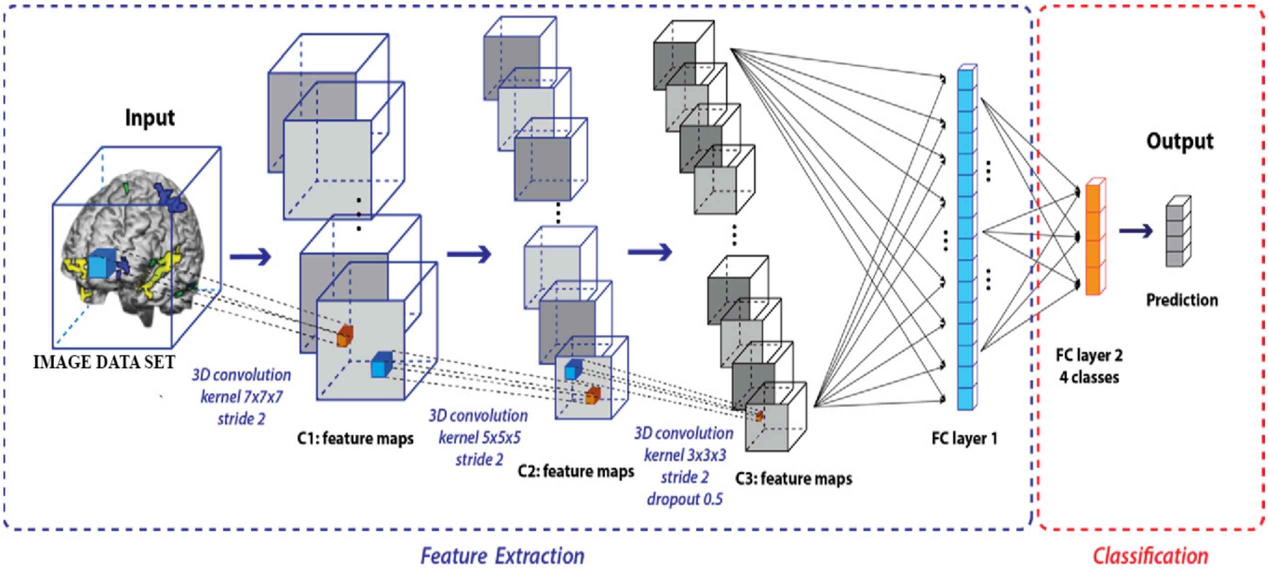


Figure 3: A 3D CNN Architecture for proposed Dataset to classify the images into different classes classification

neural network for image classification by using proposed dataset with the function of feature extraction. Given that they may be used with either a series of 2D frames or a 3D volume as input, 3D CNNs are a complicated model for computational approaches for volumetric data (e.g. slices in a CT scan). Using 3D convolution kernels and 3D pooling, methods that may be applied to volumetric data such as computed tomography (CT) images have been developed. The addition of 3D convolution kernels to the architecture increases the number of parameters, training time, and data requirements.

Training 3D CNNs on data from multiple methods is not always simple due to the limited size of medical picture datasets. Some pioneering attempts have been made along this line [23-28] to describe spectral and spatial information concurrently. 3D CNN models execute stacked convolution operations in a layer-by-layer way over spatial and spectral feature space. The generated rich feature maps are clearly the advantage of this type of 3D CNN model. These approaches, on the other hand, have three major drawbacks. To begin with, creating a more detailed 3D CNN model is tricky.

The reason for this is that as the number of 3D convolution processes rises, the solution space expands exponentially, limiting the model's depth and interpretability. Second, if a significant number of 3D convolution operations are performed, the memory cost becomes prohibitive. Third, the small size of the public hyperspectral image datasets makes it impractical to train a deeper 3D CNN model, which requires extra training instances. To address the aforementioned issues, this work proposes a unique 3D CNN model that requires only a few 3D convolution operations but produces richer feature maps [29]. Figure 3 represents the 3D convolutional neural network architecture for image classification by using proposed dataset after extracting features from that proposed dataset.

1.2. Hyperspectral imaging

Hyperspectral images (HSIs), which contain hundreds of spectral bands, are created using a network of hyperspectral imaging sensors. Since there is a very tiny wavelength gap between every two nearby bands, HSIs have a very high spectral resolution [30]. (usually 10 nm). The use of HSI analysis is widespread in a variety of industries, including materials analysis, precision agriculture, environmental monitoring, and surveillance [31–33]. The hyperspectral community's most active area of study is HSIs classification, which aims to categories every pixel in an image [34].

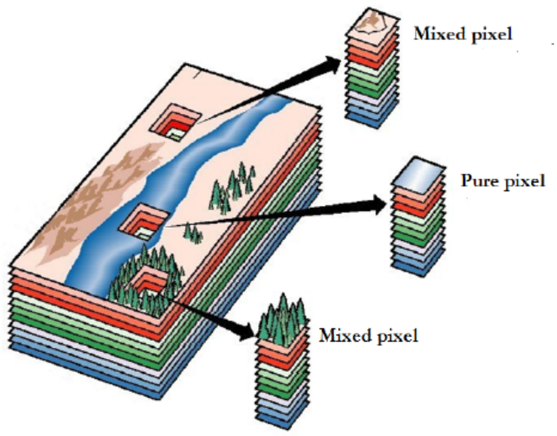


Figure 4: Hyperspectral imaging concept for classification

In figure 4 show the concept of hyperspectral imaging. The categorization of HSIs is challenging, nevertheless, due to the heavily duplicated spectral band information and few training samples [35]. In an HS image classification system, image restoration (e.g., de-noising, incomplete data restoration) [36,37], feature vectors [38], spectral un-mixing, and feature extraction [39] are all general sequential processes. Feature extraction is one of them, and it's a vital stage in HS image categorization that's been getting a lot of attention lately. A vast range of powerful hand-crafted and machine learning-based feature extraction techniques for HS image classification have been presented over the last decade [40]. These algorithms are capable of handling small-sample classification issues well. When the training size progressively expands and the training images become more complicated, they are likely to hit a performance bottleneck. This could be owing to the traditional approaches' restricted data fitting and representation abilities.

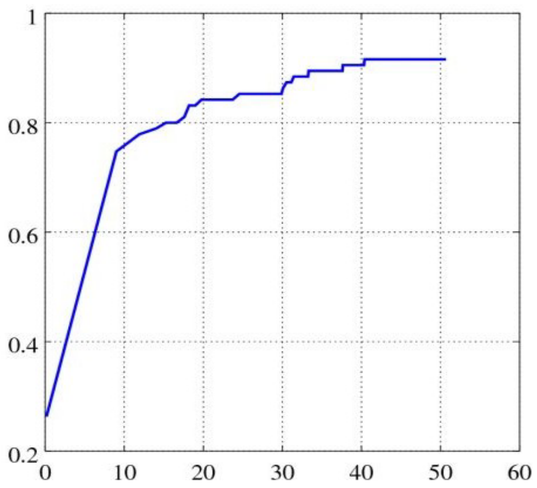


Figure 5: FROC curve for the 3D CNN

2. Related Work review

In this article they used a three-dimensional convolutional neural network to classify lung nodules in chest CT images. In this proposed method they used two techniques one is screening stage and second is discrimination stage. A CAD system's scanning stage is a standard feature. This stage narrows the initial search space and indicates a selection of the most likely candidates who should be investigated further. The screening CNN in our system is initially trained to classify 3D patches derived from each CT case using 3D convolution kernels [41]. The negative samples were chosen arbitrarily by extracting VOIs of the same size as the tested cases from a random location within the CT scan, whereas the specimens for this CNN were created by trying to extract VOIs of the same size as the tested cases from a random location inside the CT scan (in both the inside and outside of the lungs) (including both the inside and outside of the lungs). The selection of the negative patches ensured that none of the nodules would be overlapped by them. The number of negative samples obtained in this way may be nearly as big as required because the majority of the region within a chest CT is nodule-free. On the other hand, there are only a certain number of positive samples. The positive samples are reinforced by inserting flipped and rotated copies of each extracted positive patch in the training set to increase the system's invariance to small variations in nodule appearance and to decrease the aforementioned class imbalance problem. The previous section's screening stage still produces a significant percentage of false positives. The goal of the discriminating stage is to lower this number so that the clinician receives an output with high sensitivity for nodule detection and a tolerable number of false positives per case. They trained their models using a subset of 509 cases from the LIDC dataset, with slice thicknesses ranging from 1.5 mm to 3 mm, as well as an extra 25 examples for testing. One to four radiologists indicate the location of each module in the LIDC dataset, and the radiologist provides a segmentation for each newly discovered 3 mm nodule.

Only screening candidate points that pass the previously described criterion are used to evaluate the discriminatory CNN. The FROC curve for the discrimination stage is shown in Figure 4. At 15.28 FPs per case, this model achieves an 80% sensitivity. it suggests that this is accuracy is not much good as compare to other implementations of CNN, it can be improved by using other proposed methods for that we discussed is discussion session.

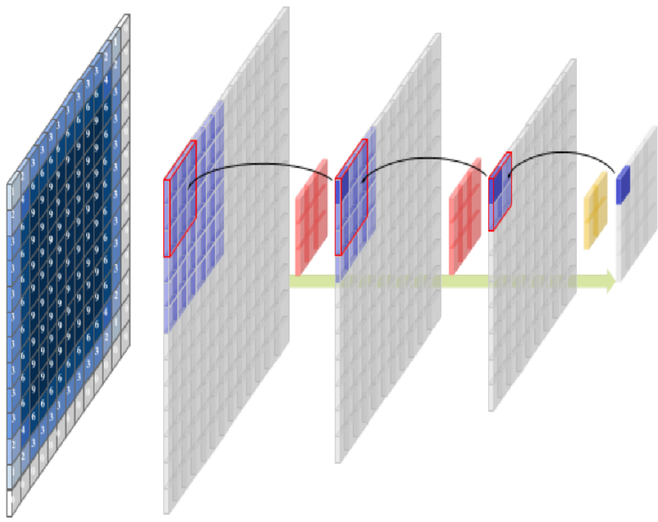


Figure 6: A convolution kernel shown graphically. The multivariate array of weights is the first section. 2D detail of a **3** × 3 kernels with stride 1 and no padding is presented in the second part.

They suggested an approach for hyperspectral image categorization that employs an adaptive convolutional neural network. Their great performance is based on the spatial linkages being exploited by convolution kernels. As a result, filter design is critical for model performance. However, there are objects of various form and orientations in hyperspectral data, prohibiting filters from seeing "all imaginable" when making decisions [42]. The deeper neurons in the visual cortex are activated by several, more complicated inputs in a hierarchical manner, whereas the output neurons are triggered by some input visual stimulus that is within their RF. On the other hand, CNNs have changed their function to mirror this behaviour. The CNN employs a single deep stack of convolutional layers, each of which defines a filter bank,

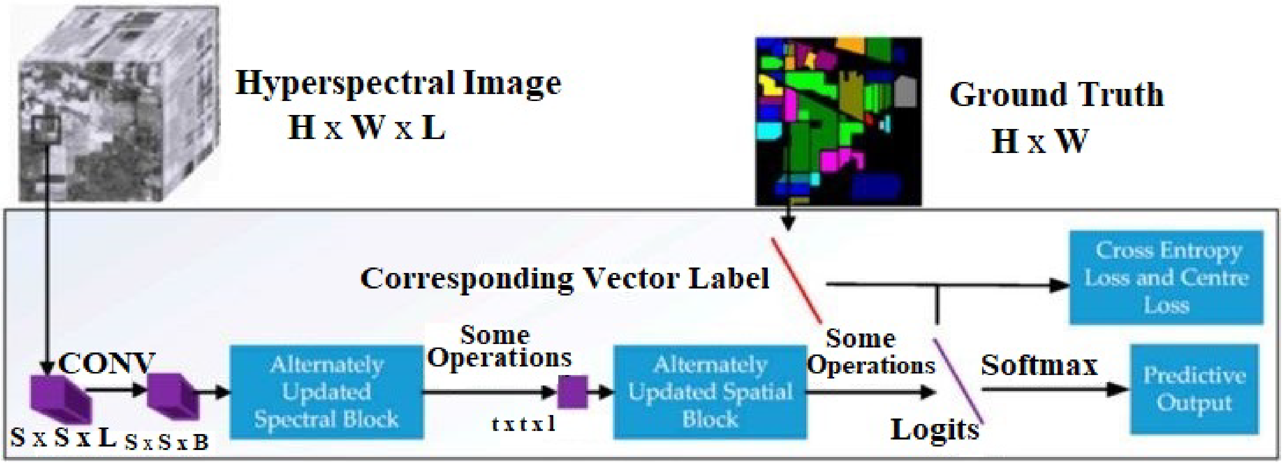


Figure 7: An overview of the proposed spectral–spatial convolutional network, which is alternatively updated from end to end (AUSSC). The convolution operation is referred known as "conv."

or a group of shareable, teachable, and locally connected weights that collectively form a linear n-dimensional kernel. A collection of data-fitted filters that sequentially traverse through the input data, overlap, and then apply themselves to the data make up the kernel.

The inputs that are included within the app's region and the filter weights are combined to create a weight value for each application. Additionally, a non-linear activation function is added to reflect the convolution layer's reaction to the search features in order to show if the features that were filtered by them are present (such as edges and forms). Following equation defines this behaviour mathematically.

$$X_j^{(\prime)} = H\left(\sum_{K \in K} X_{j+k}^{l-1} W_k^{(l)}\right) \quad (1)$$

Mathematical formula for a neural network operation in the forward propagation step. Where the superscript l indicates the Lth layer of a CNN, and in above equation H is usually implemented with ReLU and it denotes the activation function. Figure 2 illustrates this graphically. In particular, Figure 2b shows a 6 x 6 feature maps implemented with a 3 x 3 kernel with taking zero padding. In this case, k will test the input feature's locations from (2, 4) to (4, 6) (some parts of padding provided to the border of map is included) that is based on K = [1, 1] according to grid for the end coordinate j = (3, 5, z). As can be seen, the output unit only depends on the kernel's "seeing" of a small fraction of the input feature map. Any information contained in the input feature map that is outside of the RF has no bearing on the value of the output unit since this area has been designated as the RF for that unit [43].

By tracing the hierarchy back from the output feature under consideration to the input image, an effective receptive field (ERF) is established. The input data components that influence and modify the output activations are identified by the ERF. In this way, CNN's ERF resembles a Gaussian distribution, designating an area to "look at" but also exponentially concentrating attention on the centre of the feature map. The soft attention map is really based on Gaussian distributions [44,45]. One of DL's major achievements is the creation and use of the same neural architecture for the categorization of diverse pictures. The tests looked at the model's complexity, accuracy, and generalizability by counting the number of parameters. identifying and categorizing the scenes from (i)the University of Pavia and (ii) the University of Houston, two authentic, well-known HSI sceneries with a variety of spectral-spatial properties. The information is given below.

- The University of Pavia dataset [46] is an HSI picture that was taken over the university's campus in Pavia, northern Italy, in July 2002 using the ROSIS-3 airborne reflecting optics system imaging spectrometer. The picture consists of 113 wavelength channels with a frequency range of 430 to 860 nm and 610 × 340 pixels with a resolution of 1.3 m. The 42,776 tagged samples that make up the ground truth are separated into nine different land-cover classes, which include, among other urban features, asphalt, meadows, gravel, trees, steel plate, bare soil, bitumen, brickwork, and shadows.
- The lightweight tiny aerial spectrographic imager captured an HSI scene above the Houston University region for the University of Houston dataset [47] (CASI). It features 144 channels in the 380 nm to 1050 nm spectral range and 349 1905 pixels with a spatial resolution of 2.5 m. 15,029 tagged samples from 5 different courses in an urban setting are also part of the ground truth.

An innovative deep convolution-based neural network for the HSI classification process is presented in this study. The CNN classifier's effective receptive field is automatically modified by the model's deformable kernels and deformed convolutions to account for spatial deformations in HSI data from remote sensing. Instead of just being able to change the convolution, the adaptive classification network accomplishes this automatically by utilizing the distortion of the kernel itself applied to each perceptron on the input feature volume (i.e., adding an offset to the feature positions).

An upgraded spectral-spatial convolutional network has been offered as an alternate method for HSI classification. Figure 7 depicts the recommended method in broad strokes. A spatial size of S X S was selected from the raw HSI data in order to input HSI data with L channels and a size of H X W into the AUSSC network. The AUSSC picks up the spectral and spatial characteristics of an initial HSI patch using three separate convolutional kernels. The deep spectral and spatial features are modified by the alternately updated spectral and spatial blocks via recurrent feedback. The model parameters are enhanced by using the cross-entropy loss and center-loss loss functions [48]. The three 3D CNN algorithms—3D CNN, SSRN, and FDSSC—all show that a 3-dimensional edge framework outperforms 2D-CNN-based techniques and other deep learning-based approaches. This is due, among other things, to the fact that an end-to-end framework may reduce the amount of time it takes to complete a project. Reduce pre- and post-processing to ensure that the final output and original input have the closest possible relationships. Then, to increase the degree of fitness, the model is enlarged to include additional area that can be altered automatically by the data. additionally, when used with HSIs with a three-dimensional structure. In contrast to current CNN-based techniques, we offer an end-to-end CNN-based system that makes use of smaller convolutional kernels. The AUSSC employs kernels and disregards other architectures for categorizing HSI. The key distinction between the a m1 and a m2 convolutional kernels used in the 3D CNN technique is the spectral dimension. To learn spectral and spatial representations, SSRN uses spectral kernels of size 1 m and spatial kernels of size 1D, respectively. Convolutional kernels set the parameters for the model and govern which features the CNN learns. In InceptionV3, we introduce the idea of factorization into smaller convolutions [48].

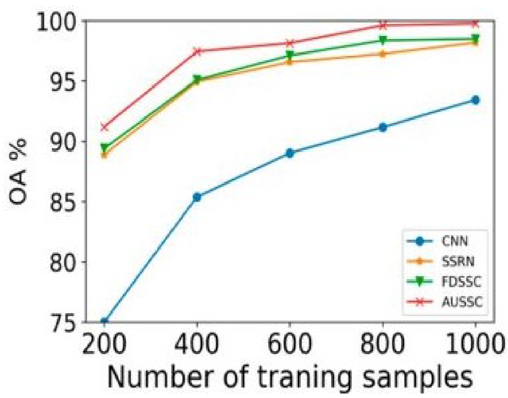


Figure 8: overall accuracy curve of above proposed methods on Houston dataset

To illustrate that the suggested strategy may decrease data reliance, they employed a very small number of training samples (200). Insufficiently labelled data is unavoidable in remote sensing applications. Furthermore, remote sensing data collection and labelling is time-consuming and costly. Therefore, creating huge, high-quality label sets is really challenging. The number of labelled samples used for learning is the most crucial variable in deep-learning supervised techniques since data dependency is one of the most critical difficulties in deep learning.

In contrast to conventional machine-learning techniques, deep learning largely depends on extensive training data to recognized possible patterns. 200 training samples are required for semi-supervised 3D-GANs as well, although their classification performance is substantially lower. [49]. Revised spectral and spatial features in HSIs were used as the fundamental building blocks to develop an end-to-end CNN-based framework for HSI classification. To learn HIS qualities and combine them into advanced features, our concurrently updated convolutional spectral-spatial network uses spatial and spectral blocks that have been modified in the opposite direction. Our technique outperforms previous deep learning-based methods by learning deeply refined spectral and spatial characteristics via alternatively updated blocks, allowing it to attain high classification accuracy.

They said that the CNN is a multilayer neural network where the convolution layer, max - pooling, and fully - connected layers are all components. The CNN model's convolution, which is the top layer, performs the convolution operation on the input data. Convolution involves performing an inner product operation on the kernel and receptive field of two matrices (learnable parameters). The feature map is constructed based on the input information and accessible features, and the kernel is often smaller than the original data and situated in the receptive field. The feature map's dimension may be effectively decreased thanks to the pooling layer. The perceptron-like convolution layer, which is composed of neurons, is multilayered, has all of the neurons linked to one another, and the output characteristics are employed in the mapping. The features are mapped into the output using this layer. Researchers found that inter band correlation has a high level of redundancy in HSI analysis. Without suffering a considerable loss of information that may be used later, the data structure of the spectral dimension can be scaled down. Contrarily, an HSI consists of hundreds of spectral bands, which makes it more difficult for the network model to handle data while also using a large amount of processing power. In recent years, PCA has been widely employed in HSI classification studies to prepare the data.

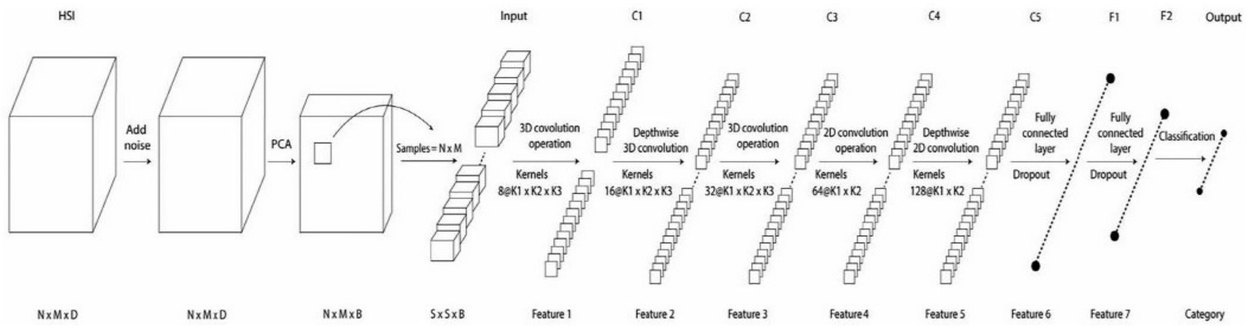


Figure 9: Hybrid convolutional neural network is used in the HSI classification architecture

In accordance with HSI classifications, the two-dimensional complexity action takes into account the input data in the spatial dimension while the three-dimensional pre-processing phase analyses the input data concurrently in the spatial and spectral dimensions. For HSIs with rich spectral information, the capacity to maintain the spectrum information of the incoming HSI data via 3-D convolution is crucial. However, whether two-dimensional convolution procedures are performed on two-dimensional or three-dimensional data, the end output is always two-dimensional, regardless of whether two-dimensional convolution techniques are employed on the HSI or not. The suggested method effectively recovers high-quality spectral and spatial feature maps from the HSI by merging the 3-D CNN and 2-D CNN. A diminishing dimension block (a Conv3-D + reshaping operation + a Conv3-D), a 3-D stacked convolution layer (a Conv3-D - fast learning layer), and ultimately a 3-D stacked convolution layer (a Conv3-D) are all used in the proposed model

From which the output feature maps are then reshaped and supplied to a Conv2-D to learn more spatial information. The output of the Conv2-layer is flattened before it is sent to the top fully connected layer. A dropout layer comes after the last fully linked layer. The proposed model's 3-D fast learning CNN block is significantly less computationally expensive and faster than the ordinary block due to the inclusion of depth-wise separable convolution and the fast convolution block in the fast learning block. To employ image classification algorithms, hyperspectral data cubes for input are split into tiny 3-D patches called PRSB, whose center pixel determines the class labels. Initially, the size of the right N m labels matched the quantity of input data patches. Although the correct labels contain a background, we transmit the data to the network as input after removing the background from the labels and patches. The convolution layer in the input image is composed of a sliding kernel. To extract important feature maps from the input, this kernel has weights that change throughout training. These qualities are used in the categorization process. The number of HSIs available is insufficient, and data is scarce. Designing a model that matches the environment is one of the hurdles in categorizing HSIs. This research provides a hybrid model of 3-D and 2-D convolution for HSI classification. To improve classification performance, spatial and spectral characteristics might be employed. In the hybrid model, the spatial-spectral information and spatial information obtained via 3-D and 2-D convolution, respectively, are integrated.

Figure 9 depicts the suggested method's design. As opposed to employing 3-D-CNN alone, combining 3-D-CNN with 2-D-CNN reduces the number of learning parameters while also using less processing power. The Adam optimizer does a better job at network optimization and cuts down on training time. In comparison to other models, the hybrid model has the best performance in terms of limiting the number of training samples and noise. We may increase the number of layers in the model and deepen the network after we have a sufficient amount of training data. Although all models have good accuracy Due to the hybrid structure's capability to exploit all of the spectral and spatial information in HSI data, the hybrid model has fewer parameters and takes less training time than the 3-D-CNN model and the 2-D-CNN model when sufficient training instances are available. Because of this, utilizing a hybrid model for HSI categorization is economical. [50] proposed a system that is implemented Artificial Neural Network for classification of FPGA cart Flower. The recommended method's superiority in the face of a short training sample and noise was confirmed by experiments on three datasets using three classification algorithms that were compared.

3. Material and Methods

The material and methods that are discussed and used in the assessed articles are based on 2D and 3D hyperspectral images and methods are mainly based on CNN. The data is collected from various sources, such as airborne or satellite sensors, and pre-processing the data to remove noise, correct atmospheric effects, and extract relevant features. Then extract relevant features from the hyperspectral data to represent the spectral-spatial information, such as using principal component analysis (PCA), independent component analysis (ICA), or texture features. The select and implement the model, in this article the main focus is on the implementation of convolutional neural network model that is mostly used for the image classification. Then train the selected deep learning model using annotated hyperspectral data, some of papers are in a supervised and some are unsupervised manner. After training evaluate the performance of the trained model using metrics such as accuracy, F1-score, precision, recall, and confusion matrix. Then implement the fine-tuning the parameters of the trained model to improve its performance, such as adjusting the regularization strength, changing the kernel function, or adding more hidden layers. The materials required for hyperspectral image classification include a computer with sufficient computational power, deep learning libraries such as Tensor-Flow or Py-Torch, and annotated hyperspectral data. Additionally, a software tool such as MATLAB or Python can be used to implement the algorithms and evaluate the performance of the models.

A systematic approach to using Convolutional Neural Networks (CNNs) would include the following steps:

- Define the problem: Determine the task you want to solve and the type of data you have available.
- Preprocess the data: Clean, normalize, and prepare the data for use in the CNN. This may include converting images to grayscale, resizing, and splitting the data into training, validation, and testing sets.
- Choose a CNN architecture: Select an appropriate CNN architecture based on the type of data you have and the task you want to solve. Common CNN architectures include Le-Net, Alex-Net, VGG-Net, Res-Net, and Inception-Net.
- Train the model: Train the model on the training data, using an optimization algorithm such as stochastic gradient descent (SGD) or Adam, and a loss function such as mean squared error (MSE) or cross-entropy.
- Validate the model: Evaluate the performance of the model on the validation data. This is used to tune the hyper-parameters of the model, such as the learning rate and batch size.
- Test the model: Evaluate the performance of the model on the test data. This provides an estimate of how well the model will perform on unseen data.
- Deploy the model: Deploy the trained model in a production environment, using a framework such as Tensor-Flow or Py-Torch.
- Monitor performance: Regularly monitor the performance of the deployed model and make improvements as necessary.

4. Results and Discussion

In this paper, we have reviewed and critically compared many supervised hyperspectral classification approaches from multiple perspectives, with a focus on the setup, speed, and automation capabilities of various algorithms. Popular approaches such as SVMs, neural networks (2D and 3D convolutional neural network), and deep approaches are among the techniques compared, which have been widely employed in the hyperspectral analysis field but have never been comprehensively investigated using a quantitative and comparative methodology. The article lies in its focus on the recent advancements in the classification of hyperspectral images using 2D and 3D convolutional neural networks (CNNs) with channel and spatial attention mechanisms. The review summarizes the current state-of-the-art methods and provides insights into the latest developments in the field, highlighting the strengths and limitations of different approaches. The key conclusion that can be drawn from this research is that no classifier consistently gives the greatest performance among the criteria under consideration (particularly from the viewpoint of classification accuracy). Different solutions, on the other hand, are dependent on the complexity of the analysis scenario (for example, the availability of training samples, processing needs, tuning parameters, and algorithm speed) as well as the application domain in question. The informative analysis of all the reviewed papers given in below table.

Table 1: Comparison of different methods for Hyperspectral Image classification

Paper Reference	Methodology	Dataset	Analysis
[42]	MCA and MLR	Hyperspectral and LIDAR data	The implementation of MCA and MLR on the mentioned data and obtained that these methods work good for LIDAR.
[43]	AUSSC and CNN	HSI datasets	Implemented the AUSSC and CNN on the mentioned datasets and observed that the Hyperspectral image classification is based on the size of convolution and size of layers in CNN.
[45]	GAN and CNN	Salinas, Indiana pines, Kennedy Space Center data	The proposed method is still need to enhance its functionality by changing the size of convolutional layers and max pooling.
[46]	mRMR and 2D-CNN	HSI datasets	The proposed method improves some major functionality of CNN that were not good in simple CNN, 2D-CNN enhance the classifier functionality of CNN.
[47]	Deep and Dense CNN	Indiana pines, Kennedy Space Center data, university of Pavia datasets	Deep and dense CNN implemented on all mentioned datasets, and found that it works with 15% labeled data but not produce efficient results.
[48]	CNN and MFL	HSI datasets	Proposed methods implemented on given datasets and elaborate that CNN works better as compare to MFL.

5. Conclusion

To advance the field of classification of hyperspectral images using 2D and 3D CNNs with channel and spatial attention, the following open research challenges and future research directions can be considered:

Finding more effective ways to exploit the rich spectral-spatial information in hyperspectral images to improve classification accuracy. Generalization to real-world scenarios: Improving the generalization of CNN models to real-world hyperspectral data, which can often be noisy and have complex background variations. Combining multiple sources of information:

Exploring the integration of other sources of information, such as elevation data or textual annotations, to improve hyperspectral image classification performance. Computational efficiency:

Developing more efficient algorithms to reduce the computational burden of hyperspectral image classification, especially for large-scale datasets. Robustness to atmospheric and illumination conditions:

Improving the robustness of CNN models to variations in atmospheric conditions and illumination, which can significantly impact the performance of hyperspectral image classification.

Semi-supervised and unsupervised learning: Investigating the potential of semi-supervised and unsupervised learning methods for hyperspectral image classification to reduce the need for large annotated datasets. Exploring the use of multi-scale information to improve the classification of hyperspectral images, such as using multi-scale convolutional filters or combining multiple CNNs with different receptive field size. This work will be enhanced by the use of other suitable methodologies for the categorization of hyperspectral pictures, such as Transformers, which will be dependent on his/her expectations and/or exploitation aims.

Conflict of Interest

The authors declare no conflict of interest.

References

[1]. J. Yin, C. Qi, Q. Chen, and J. Qu, "Spatial-Spectral Network for Hyperspectral Image Classification: A 3-D CNN and Bi-LSTM Framework," *Remote Sens.*, vol. 13, no. 12, p. 2353, 2021, doi: 10.3390/rs13122353.

[2]. H. Yan, J. Wang, L. Tang, E. Zhang, K. Yan, K. Yu, and J. Peng, "A 3D Cascaded Spectral–Spatial Element Attention Network for Hyperspectral Image Classification," *Remote Sens.*, vol. 13, no. 13, p. 2451, 2021, doi: 10.3390/rs13132451.

[3]. S. Pu, Y. Wu, and X. Sun, "Hyperspectral Image Classification with Localized Graph Convolutional Filtering," *Remote Sens.*, vol. 13, no. 3, p. 526, 2021, doi: 10.3390/rs13030526.

[4]. D. Hong, N. Yokoya, J. Chanussot, and X. X. Zhu, "An Augmented Linear Mixing Model to Address Spectral Variability for Hyperspectral Unmixing," *IEEE Trans. Image Process.*, vol. 28, pp. 1923–1938, 2019, doi: 10.1109/TIP.2018.2878958.

[5]. S. Gudmundsson, T. P. Runarsson, and S. Sigurdsson, "Support vector machines and dynamic time warping for time series," in *Proc. 2008 IEEE Int. Joint Conf. Neural Netw. (IEEE World Congr. Comput. Intell.)*, 2008, doi: 10.1109/IJCNN.2008.4634188.

[6]. Q. Wang, Z. Meng, and X. Li, "Locality adaptive discriminant analysis for spectral–spatial classification of hyperspectral images," *IEEE Geosci. Remote Sens. Lett.*, vol. 14, pp. 2077–2081, 2017, doi: 10.1109/LGRS.2017.2703814.

[7]. K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Las Vegas, NV, USA, 2016, pp. 770–778, doi: 10.1109/CVPR.2016.90.

[8]. V. Sharma, A. Diba, T. Tuytelaars, and L. Van Gool, "Hyperspectral CNN for Image Classification & Band Selection, with Application to Face Recognition," KU Leuven, ESAT, Leuven, Belgium, 2016, doi: 10.3390/pr11020435.

[9]. A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, Curran Associates, Inc., New York, NY, USA, 2012, pp. 1097–1105, doi: 10.1145/3065386.

[10]. I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, MIT Press, 2016.

[11]. S. Albawi, T. A. Mohammed, and S. Al-Zawi, "Understanding of a convolutional neural network," in *Proc. 2017 Int. Conf. Eng. Technol. (ICET)*, 2017, doi: 10.1109/ICENGTECHNOL.2017.8308186.

[12]. J. Schmidhuber, "Deep Learning in neural networks: An overview," *Neural Networks*, vol. 61, pp. 85–117, 2015, doi: 10.1016/j.neunet.2014.09.003.

[13]. B. Zhao, H. Lu, S. Chen, J. Liu, and D. Wu, "Convolutional neural networks for time series classification," *J. Syst. Eng. Electron.*, vol. 28, no. 1, pp. 162–169, 2017, doi: 10.21629/JSEE.2017.01.18.

[14]. K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2016, pp. 1026–1034, doi: 10.48550/arXiv.1502.01852.

[15]. M. A. Aslam, M. U. Sarwar, M. K. Hanif, R. Talib, and U. Khalid, "Acoustic classification using deep learning," *Int. J. Adv. Comput. Sci. Appl.*, vol. 9, no. 8, pp. 153-159, 2018, doi: 10.14569/IJACSA.2018.090820.

[16]. G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. 30th IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR 2017)*, Honolulu, HI, USA, 2017, pp. 2261–2269, doi: 10.1109/CVPR.2017.243.

[17]. J. Hu, Y. Kuang, B. Liao, L. Cao, S. Dong, and P. Li, "A multichannel 2D convolutional neural network model for task-evoked fMRI data classification," *Comput. Intell. Neurosci.*, vol. 2019, 2019, doi: 10.1155/2019/5065214.

[18]. B. Liu, X. Yu, P. Zhang, X. Tan, A. Yu, and Z. Xue, "A semi-supervised convolutional neural network for hyperspectral image classification," *Remote Sens. Lett.*, vol. 8, pp. 839–848, 2017, doi: 10.1080/2150704X.2017.1331053.

[19]. X. Yang, Y. Ye, X. Li, R. Y. Lau, X. Zhang, and X. Huang, "Hyperspectral Image Classification with Deep Learning Models," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, pp. 5408–5423, 2018, doi: 10.1109/TGRS.2018.2815613.

[20]. A. B. Hamida, A. Benoit, P. Lambert, and C. B. Amar, "3-D Deep Learning Approach for Remote Sensing Image Classification,"

IEEE Trans. Geosci. Remote Sens., vol. 56, pp. 4420–4434, 2018, doi: 10.1109/TGRS.2018.2818945.

[21]. Y. Li, H. Zhang, and Q. Shen, "Spectral–spatial classification of hyperspectral imagery with 3D convolutional neural network," *Remote Sens.*, vol. 9, p. 67, 2017, doi: 10.3390/rs9010067.

[22]. M. He, B. Li, and H. Chen, "Multi-scale 3D deep convolutional neural network for hyperspectral image classification," in *Proc. 2017 IEEE Int. Conf. Image Process. (ICIP)*, Beijing, China, 2017, pp. 3904–3908, doi: 10.1109/ICIP.2017.8297014.

[23]. Y. Luo, J. Zou, C. Yao, X. Zhao, T. Li, and G. Bai, "HSI-CNN: A Novel Convolution Neural Network for Hyperspectral Image," in *Proc. 2018 Int. Conf. Audio, Language Image Process.*, Shanghai, China, 2018, pp. 464–469, doi: 10.1109/ICALIP.2018.8455251.

[24]. J. Fan, H. L. Tan, M. Toomik, and S. Lu, "Spectral–spatial hyperspectral image classification using super-pixel-based spatial pyramid representation," in *Proc. Image Signal Process. Remote Sens. XXII*, Edinburgh, UK, 2016, vol. 10004, pp. 315–321, doi: 10.3390/rs12122033.

[25]. X. Yang et al., "Synergistic 2D/3D convolutional neural network for hyperspectral image classification," *Remote Sens.*, vol. 12, no. 12, p. 2033, 2020, doi: 10.3390/rs12122033.

[26]. Q. Wang, X. He, and X. Li, "Locality and structure regularized low rank representation for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, pp. 911–923, 2018, doi: 10.1109/TGRS.2018.2862899.

[27]. Y. Yuan, Y. Feng, and X. Lu, "Projection-Based NMF for Hyperspectral Unmixing," *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, vol. 8, pp. 2632–2643, 2015, doi: 10.1109/JSTARS.2015.2427656.

[28]. W. Li, Q. Du, and B. Zhang, "Combined sparse and collaborative representation for hyperspectral target detection," *Pattern Recognit.*, vol. 48, pp. 3904–3916, 2015, doi: 10.1016/j.patcog.2015.05.024.

[29]. B. Pan, Z. Shi, and X. Xu, "MugNet: Deep learning for hyperspectral image classification using limited samples," *ISPRS J. Photogramm. Remote Sens.*, vol. 145, pp. 108–119, 2018, doi: 10.1016/j.isprsjprs.2017.11.003.

[30]. S. Zhou, Z. Xue, and P. Du, "Semisupervised Stacked Autoencoder with Cotraining for Hyperspectral Image Classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, pp. 3813–3826, 2019, doi: 10.1109/TGRS.2018.2888485.

[31]. P. Ghamisi, J. Plaza, Y. Chen, J. Li, and A. J. Plaza, "Advanced Spectral Classifiers for Hyperspectral Images: A review," *IEEE Geosci. Remote Sens. Mag.*, vol. 5, pp. 8–32, 2017, doi: 10.1109/MGRS.2016.2616418.

[32]. W. Cao, K. Wang, G. Han, J. Yao, and A. Cichocki, "A robust PCA approach with noise structure learning and spatial–spectral low-rank modeling for hyperspectral image restoration," *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, vol. 11, no. 10, pp. 3863–3879, 2018, doi: 10.1109/JSTARS.2018.2866815.

[33]. D. Hong, N. Yokoya, J. Chanussot, J. Xu, and X. X. Zhu, "Joint and progressive subspace analysis (jpsa) with spatial-spectral manifold alignment for semi-supervised hyperspectral dimensionality reduction," *IEEE Trans. Cybern.*, vol. 51, no. 7, pp. 3602–3615, 2021, doi: 10.1109/TCYB.2020.3028931.

[34]. F. Luo, T. Guo, Z. Lin, J. Ren, and X. Zhou, "Semisupervised hypergraph discriminant learning for dimensionality reduction of hyperspectral image," *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, vol. 13, pp. 4242–4256, 2020, doi: 10.1109/JSTARS.2020.3011431.

[35]. D. Hong, X. Wu, P. Ghamisi, J. Chanussot, N. Yokoya, and X. X. Zhu, "Invariant attribute profiles: A spatial-frequency joint feature extractor for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 6, pp. 3791–3808, 2020, doi: 10.1109/TGRS.2019.2957251.

[36]. B. Rasti et al., "Feature extraction for hyperspectral imagery: The evolution from shallow to deep: Overview and toolbox," *IEEE Geosci. Remote Sens. Mag.*, vol. 8, no. 4, pp. 60–88, 2020, doi: 10.1109/MGRS.2020.2979764.

[37]. S. Hamidian, B. Sahiner, N. Petrick, and A. Pezeshk, "3D convolutional neural network for automatic detection of lung nodules in chest CT," in *Medical Imaging 2017: Computer-Aided Diagnosis*, vol. 10134, p. 1013409, 2017, doi: 10.1117/12.2255795.

[38]. M. E. Paoletti and J. M. Haut, "Adaptable Convolutional Network for Hyperspectral Image Classification," *Remote Sens.*, vol. 13, no. 18, p. 3637, 2021, doi: 10.3390/rs13183637.

[39]. H. Le and A. Borji, "What are the receptive, effective receptive, and projective fields of neurons in convolutional neural networks?," *arXiv*, 2017, doi: 10.48550/arXiv.1705.07049.

[40]. H. Gao, X. Zhu, S. Lin, and J. Dai, "Deformable Kernels: Adapting Effective Receptive Fields for Object Deformation," *arXiv*, 2019, doi: 10.48550/arXiv.1910.02940.

[41]. A. Araujo, W. Norris, and J. Sim, "Computing receptive fields of convolutional neural networks," *Distill*, vol. 4, p. e21, 2019, doi: 10.23915/distill.00021.

[42]. X. Xu, J. Li, and A. Plaza, "Fusion of hyperspectral and LiDAR data using morphological component analysis," in *2016 IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, 2016, pp. 3575-3578, doi: 10.1109/IGARSS.2016.7729926.

[43]. W. Wang, S. Dou, and S. Wang, "Alternately updated spectral–spatial convolution network for the classification of hyperspectral images," *Remote Sens.*, vol. 11, no. 15, p. 1794, 2019, doi: 10.3390/rs11151794.

[44]. C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 2818-2826, doi: 10.48550/arXiv.1512.00567.

[45]. L. Zhu, Y. Chen, P. Ghamisi, and J. A. Benediktsson, "Generative adversarial networks for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 9, pp. 5046-5063, 2018, doi: 10.1109/TGRS.2018.2805286.

[46]. [46] S. Ghaderizadeh, D. Abbasi-Moghadam, A. Sharifi, N. Zhao, and A. Tariq, "Hyperspectral image classification using a hybrid 3D-2D convolutional neural networks," *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, vol. 14, pp. 7570-7588, 2021, doi: 10.1109/JSTARS.2021.3099118.

[47]. M. Paoletti, J. Haut, J. Plaza, and A. Plaza, "Deep&dense convolutional neural network for hyperspectral image classification," *Remote Sens.*, vol. 10, p. 1454, 2018, doi: 10.3390/rs10091454.

[48]. Q. Gao, S. Lim, and X. Jia, "Hyperspectral image classification using convolutional neural networks and multiple feature learning," *Remote Sens.*, vol. 10, no. 2, p. 299, 2018, doi: 10.3390/rs10020299.

[49]. U. Farooq and R. B. Bass, "Frequency Event Detection and Mitigation in Power Systems: A Systematic Literature Review," *IEEE Access*, vol. 10, 2022, doi: 10.1109/ACCESS.2022.3180349.

[50]. M. F. Ahmad et al., "Tracking system using artificial neural network for FPGA cart follower," *J. Phys. Conf. Ser.*, vol. 1874, no. 1, 2021, doi: 10.1088/1742-6596/1874/1/01.