Analysis Of Minimum Spanning Tree Algorithms

[1]Parth Solanki, [2]Rushirajsinh Gohil, [3]Jenil Patel, [4]Ekta Unagar, [5]Dhaval Chandarana

[1,2,3,4,5]Dept. Of Information Technology

[1,2,3,4,5]Gyanmanjari Institute Of Technology Bhavnagar,India

[1]solankiparth41252@gmail.com, [2]gohilrushirajsinh1872@gmail.com,

[3]jenilpatel614@gmail.com, [4]ehunagar@gmiu.edu.in, [5]drchandarana@gmiu.edu.in

Abstract-The minimum spanning tree (MST) Problem is an excellent way to show how Graph Theory and Network Optimization relate to each other, as well as how many of the classic algorithms are still in play today. As pointed out by Pettie & Ramachandran (2003), the History of the MST Problem and its Optimality has been expanded upon through their work and since then many researchers have focused on creating more Data and other types of Data to explore the original question of whether the MST is indeed Optimal and provides a theoretical path for further Research/Field Advancements, as well as to provide Data Types, Data Representation and eventually the Bioinformatic approach using MST Modelings. Many Articles, etc from 2010, 2020 and 2024 are more references of, and had provided ways to inform larger Data, Data Types and Data Representations. In addition, there are more references to larger Data, Data Type representations using visualized representation with Bioinformatic methods within the Literature. This paper will provide an overview of the Literature developments, complications and Performance of MST and the Algorithm(s) with a focus on Scalability, Adaptivity (to larger), AI and materials of a larger scale.

Index Terms—Minimum Spanning Tree (MST), Kruskal's Algorithm, Prim's Algorithm, Dual-Tree Borůvka, Approximate MST, Data Clustering, Network Optimization, Big Data Analysis, Computational Efficiency.

I. INTRODUCTION

Minimum-Spanning Tree Problems are very important problems faced by graph theorists and the Computer Science community. As a general principle, the purpose of Minimum Spanning Trees

is to connect every single vertex to every other vertex with edges of the fewest number possible while avoiding creating any cycles and minimizing the sum of the weights on all edges. While this goal seems like a very straightforward and simple objective, in fact Minimum Spanning Trees represent an extremely useful idea with hundreds of applications in networking, optimization, communication systems, etc. The first Minimum Spanning Tree algorithms were created in the first decade of the 1900's for optimizing the distribution of electricity over electrical grid networks; they have since evolved into highly effective general-purpose Minimum Spanning Tree algorithms for connecting arbitrary graphs.

Over the years there have been multiple algorithmic discoveries that have been found to perform very similarly to Minimum Spanning Tree algorithms. Some of these will be classified as some of the most effective algorithms for Minimum Spanning Trees with varying amounts of success. Many of the original Minimum Spanning Tree algorithms were Borůvka's, Kruskal's and Prim's and they all were developed during a time when computer systems were being designed to be as efficient as possible and thus, as writers moved to make and share general-purpose algorithms, bore witness to a burgeoning concept of general-purpose algorithms.

II. LITERATURE SURVEY

Over time, many iterations and designs on Minimum Spanning Tree (MST) Algorithms were made starting from Greedy Algorithms to Adaptive, Approximate, and High Dimensional MST versions. The earliest example of a systematic approach to determining an MST is Borůvka's Algorithm [1], which focused on reducing the costs associated with the design of a wire harness through Parallel Edge Selection. Kruskal's [2] implementation of the Union-Find algorithm in conjunction with Edge Sorting improved Borůvka's algorithm in the sense that it was designed with Sparse Graphs in mind. An Incremental Algorithm Based on Vertices was developed by Prim [3], improving upon the performance of Kruskal's and Borůvka's Algorithms for Dense Graphs. A recent evaluation of the Classical Algorithms conducted by Ayegba et al. [4] confirmed that the performance of Classical Algorithms varies based on Topology; this property continues to be relevant to today's Computational Methods.

In terms of the theoretical upper limits of performance on MSTs as applied to the Decision Tree model, the work of Pettie and Ramachandran [5] shows a step further than other existing MST Algorithms via their application of a Bounding Error with Soft Heaps. Furthermore, they were able to show a direct relationship between Approximate and Deterministic Optimization of MST Algorithms.

Finally, March et al. [6] produced the Dual-Tree
Borůvka MST Algorithm through the use of kd-Trees for the purpose of partitioning Space to accelerate the Distance Calculations necessary to compute an MST. By eliminating the need for Unnecessary Distance Calculations for cases where the Data Points are Spatially Separable, the

Dual-Tree Borůvka MST Algorithm can Calculate the Distance Calculations required for the MST faster than other MST Algorithms.

The Adaptive Mini-MST (AMST) framework created by Li et al. [8] provides a means to identify aberrancies by forming several localized mini-MSTs while utilizing adaptive thresholds. This means that AMST can create self-adjusting and threshold dependent (parameter contingent) methods to identify outliers in both medical and finance domains. Almansoori et al. [9],[10] have created the Approximate MST (AMST) to show that it has almost linear time complexity ($O(n^{1.07})$), which comes at a slight (<6%) reduction in accuracy, making it very well suited for use with applications that require large cluster sizes and those that leverage Artificial Intelligence. These papers, taken together, show a continuing evolution in the MST paradigm: from classical greedy optimization [1] – [4], to theoretical optimal serialization [5], to now scalable, domain specific, and adaptive MST computation for use across data analysis, network optimization, and AI systems [6] – [10]. Thus, the transition that has been documented in the papers makes explicit that past papers have been primarily method-based, while continuing to describe some of the underlying archetypes of MST as new ways of developing techniques to be used for data analysis, network optimization, and AI systems.

A. CLASSICAL MST ALGORITHMS

At the turn of the 20th century, Minimum Spanning Tree (MST) algorithms were designed to provide effective solutions to communications and electrical network problems. The main historical and conceptual basis of MST research is derived from the three classical algorithms, Borůvka's Algorithm (1926), Kruskal's Algorithm, and Prim's Algorithm. Although all three algorithms employ a greedy method for edge selection and all build the same spanning tree, there are differences in how each selects its edges and the data structures used to construct the spanning tree [1], [7].

(1) Borůvka's Algorithm (1926)

The first systematic MST algorithm was created by Otakar Borůvka when he was designing electrical distribution networks in Moravia [1]. In Borůvka's method, each vertex is treated as an independent component, and in each iteration of the algorithm, the minimum outgoing edge associated with a component is determined. The minimum outgoing edges for all components are then used to join those components together. This process is repeated until there is a single spanning tree containing all vertices.Borůvka's algorithm has a strong tendency to parallelism and is, therefore, ideally suited for parallel and distributed computing, as it allows for simultaneous edge selection and does not require cycles. Borůvka's algorithm reduces the number of components by half with each iteration; therefore, its time complexity is O(E log V) [7]. Borůvka's method also served as a pre-process for large-scale distributed computations and laid a foundation for subsequent work in this area according to Ayegba et al. [7].

(2) Kruskal's Algorithm (1956)

Joseph B. Kruskal developed a very simple greedy algorithm for finding minimum-weight edges connecting two parts of an unconnected graph using edges, which means constantly choosing the smallest edge until you connect two previously unconnected parts of a graph [2]. He achieved this by sorting edges in increasing order. The Union-Find (Disjoint Set Union) datastructure can efficiently assist in the detection of cycles.

The time complexity for sorting edges is O(E log E). Since Kruskal's algorithm does not require checking for collisions and thus has a lower overhead when it comes to dense graphs (having fewer edges compared to vertices), it is well-suited to sparse graphs as described by Ayegba et al. [7].

Because of its relatively easy implementation and low memory footprint, Kruskal's algorithm is frequently used in network design and clustering as stated by Ayegba et al. [7].

(3) *Prim's Algorithm (1957)*

A good analogy for Prim's algorithm is Dijkstra's shortest path algorithm because they share similar principles but differ in their approach; on the one hand, using a vertex-based incremental method and the other, using a greedy algorithm based on edges.[3]. Prim uses an arbitrary starting point and incrementally builds a solution by adding the minimum edge between an unconnected vertex and an included vertex until all vertices in the solution tree are connected to the tree.The time complexity for implementing Prim's algorithm using a binary heap with an adjacency list is O(E + V log V) [7].When iterating on dense graphs that have a number of edges near to V2, it is empirically proven that running Prim's algorithm from every vertex will be more efficient than simply finding the minimum-weight edges connecting two unconnected parts using Kruskal's algorithm.

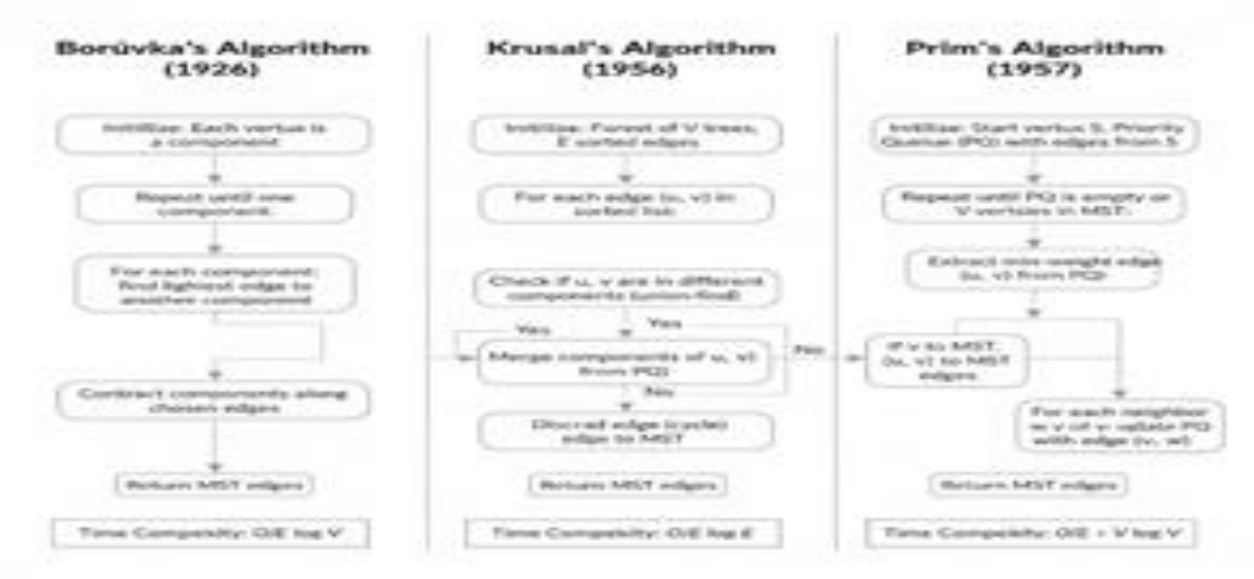

Fig. 1. Overview of classical mst algorithms.

B. THEORETICAL DEVELOPMENTS (PETTIE & RAMACHANDRAN, 2002)

In 2002, Pettie and Ramachandran significantly advanced the minimum spanning tree (MST) algorithms on a theoretical level. They proved that there is an optimal MST algorithm for the decision tree model (as referenced in the paper). The algorithms in Chazelle (2000) were the most sophisticated cutting-edge MST algorithms available at that time, providing high-quality solution times of around O (m α (m, n)) but they were not practical because of their complexity. Pettie and Ramachandran framed the MST construction problem as a theoretical computer science problem, and through this framing, provided an answer to the long unanswered question regarding the optimal complexity for constructing an MST.

(1) Theoretical Motivation

Decision-tree complexity is defined mathematically as T * (n,m), where T * (n,m) is the least number of comparisons of edges, ranked by their weight, to find the minimum spanning tree (MST) for a graph that has n nodes and m edges. The authors of the paper [4] presented T * (n,m) as the basis for calculating the time required to find a minimum spanning tree using any algorithm.

Mathematically, T * (n,m) cannot be expressed using one equation, but T * (n,m) can be expressed as an upper bound of O (m*α (n,m)) and a lower bound of Ω (m) for the theoretical gap in the time required to find minimum spanning trees. As a result, T * (n,m) defines two upper and lower bounds of theoretical time complexity to find the minimum spanning tree. The authors of the paper demonstrated that an almost optimal MST algorithm currently exists, even though a complete analysis of the time complexity and running time of such an algorithm cannot be provided at this time. Such a proof makes it possible to provide a theoretical foundation for future algorithms to calculate minimum spanning trees that will conform to these bounds on time complexity.

(2) Algorithmic Design and Key Techniques

At In their original paper on Minimum Spanning Trees (MSTs), the authors introduced the idea of decision-tree complexity or T * (m, n), which denotes the lowest number of comparisons of edge weights required to create the MST of a graph with m edges and n vertices [4]. Pettie and Ramachandran showed that no algorithm could produce a bound asymptotically larger than T* (m, n), by demonstrating that MSTs could be constructed using a running time of O(T * (m, n)).

A hybrid Minimum Spanning Tree (MST) algorithm was developed by the authors, using an iterative procedure to execute Borůvka's phases of component merging. A Soft Heap allows for quick selection of the edges at each step of the iteration. The iterative process finds the lightest edge at each iteration, although it does not guarantee finding them in the correct order; rather, it optimizes speed, at the expense of some slight inaccuracy. Each merge operation is performed recursively; therefore, eventually, the edges will be added back into their proper places if an edge resulted from an incorrect merge. A correct MST will still be produced through this process, but will have required fewer comparisons, demonstrating an improved efficiency for amortized costs, and will still provide for a theoretically optimal solution [4]

(3) Theoretical Importance

While the research conducted by Pettie and Ramachandran had more of a theoretical nature than a practical application, what they did provide was the first proof that the complexity of decision trees for constructing minimum spanning trees (MST) in terms of algorithms is optimal. Thus, this research has provided a means of addressing the theoretical question of what is meant by "MST" on the model of computation.

Additionally, this research offered innovative new methods to explore the relationship between data structures and optimal algorithms. Research that builds on this idea has been conducted on approximate and streaming MST algorithms by March et al. (2010) [5], Li et al. (2023) [8], and Almansoori et al. (2024) [9]. This research was stimulated by the realisation of Pettie and Ramachandran that introducing a small amount of inaccuracy, as achieved through their use of the Soft Heaps data structure, allowed for much better performance in these algorithms.

(4) The Legacy of Classical Techniques and Recent Advances

The Pettie-Ramachandran methodology differs from classical approaches because, although Kruskal's and Prim's solutions present an elegant implementation on the basis of their design, neither solution has been proven theoretically optimal in terms of Edge-Weight Comparisons [2],[3],[7]. The research by Pettie and Ramachandran indicates that the Asymptotic Behaviors of Classical Approaches (which depend on node comparisons) are comparatively slower than the current equivalent on the Decision-Tree that was developed in the studies performed by Kruskal and Prim.However, despite these findings, some of the influence of these researchers' work is found within the more modern works of Contemporary Algorithms.For instance, the AMST [9] and Mini-MST [8] are both examples of finding a compromise between the time spent on Computing to produce the Minimum Spanning Tree accurately and the ability to produce an accurate Minimum Spanning Tree with creative methodologies such as the Dual-Tree Borůvka Algorithm [5] and the Tree Mapping Algorithm (TMAP) [6]. Thus, the Pettie-Ramachandran model provides a connection back over the last century, from the traditional theories of Borůvka and the research about Minimum Spanning Trees to the most sophisticated Adaptive Big Data algorithms. It serves as a foundation for continuing future research efforts.

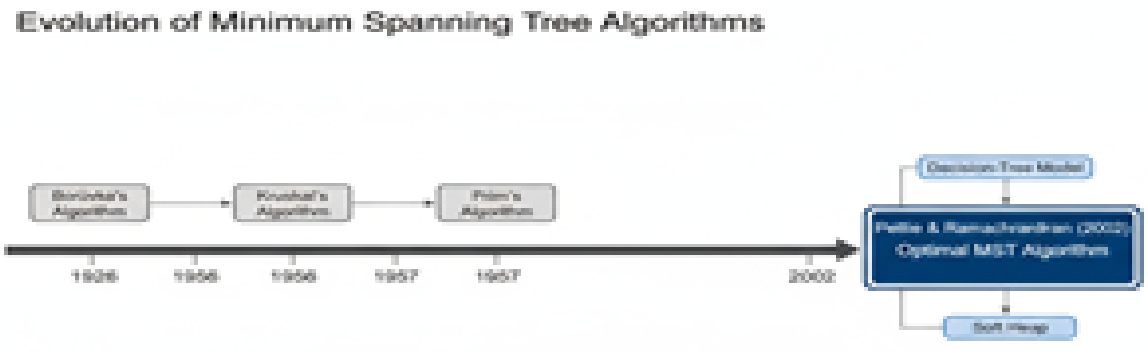

Fig. 2. Timeline-style concept diagram.

C. CURRENT ALTERNATIVES TO MST

Minimum Spanning Tree Algorithms (MST) have been rigorously validated through scientific methods. However, as dataset sizes, dimensions, and complexities increase, MST-based algorithms have struggled to perform optimally in large scale production type applications. As a result, more recent versions of MST-based algorithms emphasise approximating solutions, enabling larger datasets to be handled, improving scalability, and enhancing flexibility to accommodate larger datasets. These aspects of approximate, scalable, and flexible approaches can also be utilised to provide additional context to developing big data analytics, bioinformatics, and the visualization of data [5]-[9].

(1) March et al. (2010) Dual-Tree Borùvka Algorithm

First presented by March, Ram, and Gray in 2010 [5], the Dual-Tree Borùvka algorithm represented the first significant breakthrough in fast MST computation in both Euclidean and high-dimensional spaces. The algorithm changed Borůvka's initial concept of selecting edges in parallel by establishing a dual-tree traversal and the spatial partitioning data structure (kd-trees).

The Hierarchical structure of hierarchical clustering enables a significant reduction in the number of distance comparisons to find nearest neighbor pairs by reducing the total number of distance computations required (with a maximum computational cost of O (N log N)), due to the necessary information contained within the clusters. Furthermore, due to the fact that the architecture is built on dual trees, there are a lot more distances/points that can be calculated at once as opposed to each pairwise combination using the traditional approach.

Dual-Tree Borůvka can provide real-world applications for fields such as image segmentation, and Spatial Clustering and astronomy, where millions of individual scattered points exist in the datasets that are currently available to researchers. Due to its ability to quickly and accurately calculate large volumes of geometric data, the algorithm has great value in both computational geometry and data mining.

(2) Approximate Minimum Spanning Tree (AMST) Almansoori and colleagues (2024)

As Probst and Reymond (2020) began calling MST computation a visualization task instead of an optimization task, it was noted as a key redirect in MST research (6). In order to create a two-dimensional tree layout, the TMAP algorithm (Tree Map of High-Dimensional Data) uses a traditional Kruskal Minimum Spanning Tree (MST) process combined with approximate locality-sensitive hashing (LSH) to form a k-nearest neighbour graph.

A tree-like structure created from TMAP allows users to see millions of chemical molecules, genomic sequences, or other types of data (text) on their personal computers and still keep track of pairs within the same family in addition to pairs from different families, so that all important relationships are maintained. TMAP can also function as an interactive means of visual exploration for cheminformatics, bioinformatics and semantic data mining since the algorithm operates very efficiently (O (N log N)) as well as scales well [6]. This demonstrates how a minimum spanning

tree (MST) can be adapted for human-centered data interpretation by changing the MST optimization process into a way of representing structural relationships visually.

(3) Approximate Minimum Spanning Trees: Almansoori et al. (2024)

An Approximate Minimum Spanning Trees algorithm was developed by Almansoori, Meszaross, and Telek (2024) for achieving Near Linear Scalability of calculating Minimum Spanning Trees costs using very large datasets. The AMST specific approximates the Minimum Spanning Tree cost, and it's also the equivalent amount of work to find additional SCM global optimisation techniques using a combination of sum graphs and crawls, or also called SBC, in five to six percent of the time, respectively.

The algorithm allows for MST construction on data with billions of elements and has a low memory footprint and runs in O(n^1.07) time. While the AMST is an approximation of the MST, the features of its structure make it an especially effective method for real time graph analysis, AI-based analysis, and big data clustering. [9]

Additionally, the AMST has a very favourable trade off for efficiency in terms of accuracy and memory for large-scale or memory-constrained situations, making AMST a good alternative to the traditional exact MST algorithms.

(4) Adaptive Mini-MST (MMOD) — Li et al. (2023)

Li and colleagues (2023) have proposed an Adaptive Mini-MST (MMOD) framework to identify outliers and anomalies for financial and biological data [8]. The MMOD Framework creates a number of mini-mst cascaded to each other instead of a single MST (mst) as in traditional methods using the dynamic measure of density of node locations through a series of iterations and a defined threshold, which generates a one-time only tree from all nodes based on the defined threshold. For the grouping of the nodes, the criteria for choosing which nodes to group together are based on the density of the nodes in relation to their surrounding cluster. The criteria for grouping are the highest density or cluster and the lowest or vacant area of the node cluster as the corresponding density. Therefore, the density grouping of nodes is done automatically without any intervention from the user and is determined by the data distribution and results from the analysis.

The MMOD algorithm allows for a significant reduction in the number of operations performed over a data set and is especially useful at a time when people are producing and interacting with large volumes of dynamic information - if not outright impossible - using standard methods for dynamic dataset analysis.

The MMOD algorithm represents a combination of established graph theory and recent advances in the area of adaptive or learning (ML) algorithms; specifically adding an additional dimension to MSTs, or minimum spanning trees.

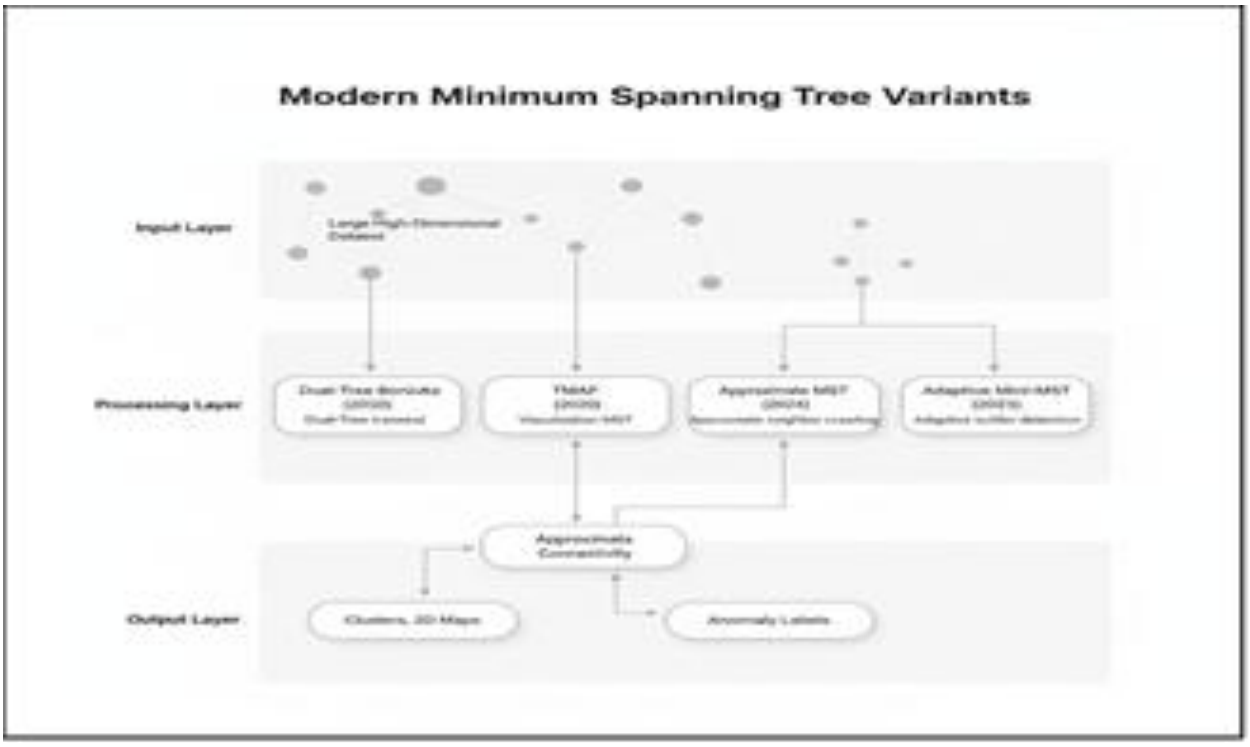

Fig. 3. Layered system diagram.

D. COMPARATIVE ANALYSIS OF ALGORITHMS

The evolution of Minimum Spanning Tree Ant algorithm also demonstrates a continuing desire to combine an efficient use of memory, time, and user-friendly. Each algorithm class has a different focus regarding priority areas, classical methods, theoretical methods, and contemporary methods with contemporary methods being developed with more of a focus on scalability, versatility and domain specific. In contrast, historical, or classical, development focuses on ease of use and correctness while theoretical development focuses on achieving optimal complexity with minimum computational resources [1], [4], [5], [7] – [9].

(1) The Comparison Work

There are multiple criteria that are commonly used when comparing MST algorithms [7]:

Computational Complexity: Performance is assessed using asymptotic notation (O-notation).
Memory Requirements: This evaluates the viability of the algorithms on large datasets.
Data Suitability: This assesses how well-suited an algorithm is to working with high-, sparse-, or dense-dimensional graphs.
Scalability: This evaluates how the algorithms will perform as the graphs get larger and larger.
Accuracy Versus Approximation: This determines whether or not the MST produced by an algorithm is exact or nearly exact.
Practical Utilization of an MST Algorithm: This evaluates the use of each algorithm in actual practice.

These criteria provide a standardized method for evaluating the capabilities of the different historical and contemporary MSC algorithms.

(2) Classical Algorithms (Borůvka, Kruskal, Prim)

The classic algorithms Borůvka [1], Kruskal [2] and Prim [3] are based on the concept of graph optimization. Most of the current work on parallel minimum spanning tree (MST) algorithms can be attributed to the original model proposed by Borůvka in 1926, which approximates the globally optimal tree by selecting and combining edges in parallel to form a tree or network from local minima. Kruskal's algorithm, introduced in 1956, applied a global greedy approach to building the minimum spanning tree and achieved a time complexity of O(E log E) through the use of a union-find data structure to sort edges. Although Kruskal's algorithm provides a way of constructing the MST for sparse graphs, such as communications and transportation networks, it can be used for dense graphs as well using Prim's algorithm (a vertex-extended method) and binary heaps. In addition, while Kruskal's algorithm provides the most memory-efficient method for constructing the minimum spanning tree for sparse networks, Prim's algorithm is the best approach for dense connectivity Ayegba et al. [7]. As Borůvka's greedy nature continues to inspire many distributed and GPU-based MST frameworks in modern computing, all three classical algorithms remain fundamental to finding a compromise between accuracy and computational feasibility for MST research.

(3) Progress in Theory (Petty + Ramachandran 2002)

The work of Pettie + Ramachandran constituted a break with past approaches in MST literature. Up until that time MST literature had generally concentrated on developing algorithms, with a small percentage studying the theoretical optimality of MST's, while the work of Pettie + Ramachandran demonstrated that it was possible to construct an optimal procedure for finding an MST in the decision tree model of computation. Additionally, they provided a theoretical upper-bound for the efficiency of any algorithm for finding an MST. An additional important aspect of the method proposed by Pettie + Ramachandran was that it was based on merging edges together in a manner similar to Borůvka, using the Soft Heap as the data structure. The asymptotic upper bound for the computational time of their proposed algorithm would be O(T*(m,n)), which would provide the best possible upper-bound for the asymptotic efficiency of any MST algorithm. Therefore, although the computational complexity of the proposed procedure is extremely high, and it is not intended to be implemented, it will remain the mathematical standard for evaluating any future proposed MST algorithms.

Almansoori et al. drew attention to the progressive evolution of static guaranteed correctness to dynamic adaptive guaranteed correctness and asserted that many of the key philosophical underpinnings of approximate and streaming algorithms are derived from the concept of bounded inaccuracy and amortised efficiency as defined by Pettie + Ramachandran.

(4) Modern Variants (2010–2024)

Recent developments in Minimum Spanning Tree (MST) algorithms, such as the 2010 introduction of the Dual-Tree Boruvka method [5], allow for both efficient management of extremely large datasets and an increase in the level of dimensionality and complexity associated with MSTs, while still maintaining reasonable, interpretable, and computable time performance by running on state-of-the-art computational capabilities. In 2006, researchers discovered that the development of the Dual-Tree Borvka Method was a major turning point; it included a new approach to the representation of spatial data that allowed for the execution of MST algorithms in O(N log N) time. As a result of the introduction of this type of structure, it was possible to process distances computed between points more efficiently. Researchers who used the Dual-Tree Borvka method to solve geometric and/or spatial problems frequently cited its advantages for use in working with large datasets. The TMAP Framework [6] was another major example; the TMAP Framework defined a new way of working with MST's for visualizing millions of points in both high- and low-dimensional spaces (e.g., 2D) as a way to improve the interpretability of molecular spatial data while conducting cheminformatics or bioinformatics research. This work demonstrates the various opportunities that exist for researchers who are working with very large datasets, specifically with respect to molecular and genomic datasets.

The Approximate Minimum Spanning Tree (AMST) algorithm [9] exhibits improved scalability over Minimum Spanning Trees (MSTs) with an upper bound time complexity of almost $O(n^{1\cdot07})$ and only a small edge case on accuracy ($\leq$6%), thus exhibiting AMST as a highly scalable clustering algorithm for large data, real-time processing, and artificial intelligence. The Adaptive Mini-MST (MMOD) algorithm [8] builds upon the same foundational framework but incorporates dynamic thresholding and local learning approaches to allow for self-adjustment while identifying and correcting for anomalies within domain-specific datasets such as high-dimensional medical and financial data. In conclusion, the development of these new algorithms represents a paradigm shift away from conventional MST computations, thus increasing to the level of intelligent analytics based upon application rather than traditional computation methods and preserving the robustness of MSTs into modern-day data analytics and decision-making models.

(5) Modern Variants (2010–2024)

MST has conceptually progressed from pure optimization towards adaptive, approximate, and contextual computation. The Dual-Tree Borůvka paper [5] focuses on geometry and how geometric data can scale spatially. The TMAP paper [6] transformed the way we visualize and create visualization methods for large and high dimensional datasets. The AMST paper [9] connects the theoretical with the practical with the concept of a near-linear time approximation of MST. Finally, MMOD [8] also provides a link to intelligent anomaly detection and adaptive learning through MST reasoning. All four of these papers are grounded in the original concepts of edge minimization and component merging as described in Borůvka's original papers [1]. As such, they continue to build on the original concepts of MST and adapt them to today's challenges associated with big data, visual analytics, and intelligent computation.

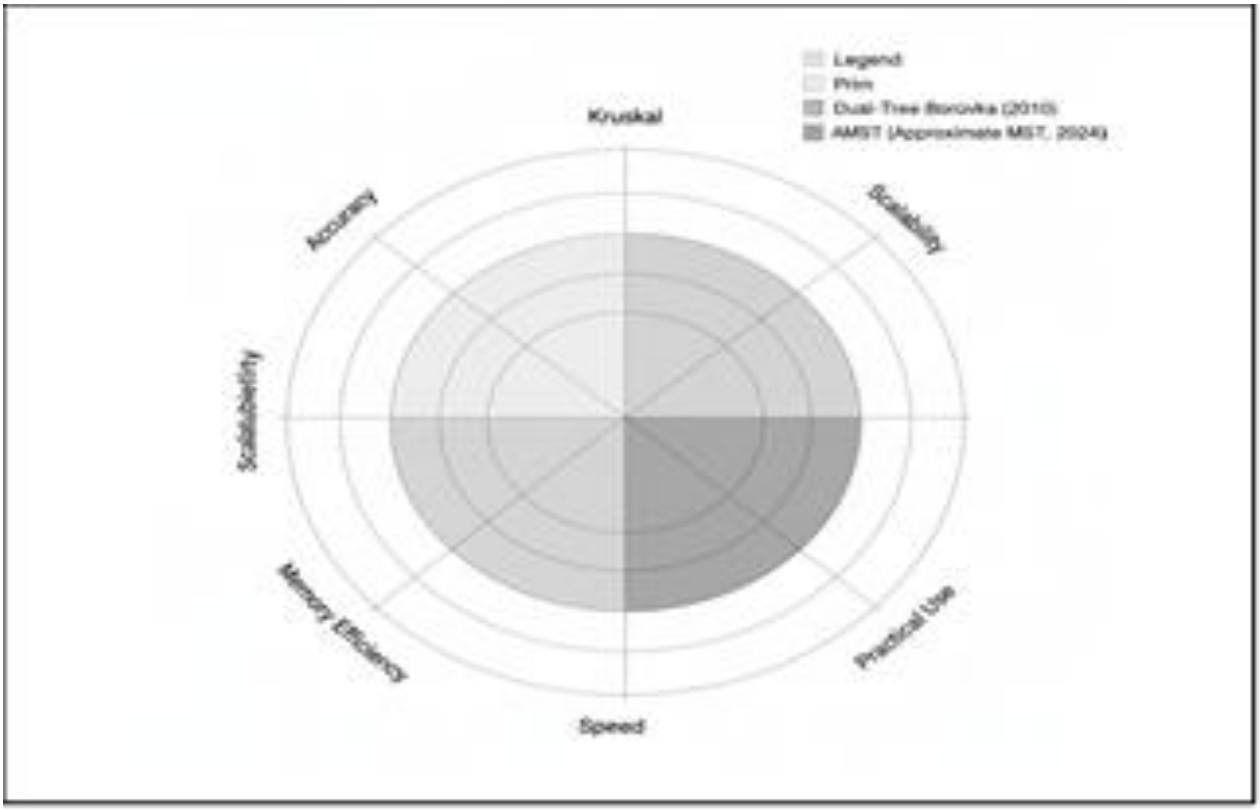

Fig. 4. Radar (spider) chart comparing.

E. APPLICATIONS ACROSS DOMAINS

The Minimum Spanning Tree (MST) is widely used and explored in many different areas of Computer and Engineering Sciences. Although the MST is developed primarily for the approximation of electrical networks, the techniques used to construct the MST have become increasingly useful across a variety of fields in the forms of various data structuring systems. Thus, there is growing interest in how the MST can be used to create efficiency and reduce costs associated with establishing connectivity between various points or objects.

The MST is an example of how an efficient structure can represent a relationship between objects without the need for additional cost information on how that relationship was formed. This is important, as many datasets can contain a significant amount of redundant, misplaced, or unstructured data. The MST has also been utilized in various areas of Engineering, Data Science, Bioinformatics, Visualization, Anomaly Detection, and beyond. The continued development of MST algorithms by Borůvka (1926), Kruskal (1956), Prim (1957), Dual-Tree Borůvka (2008), TMAP (2014), AMST (2016), and MMOD (2019) represents the continued use and growth of the MST and the associated technologies and data environments.

(1) Designing and Optimizing Networks

The primary, and most apparent, purpose for multi-hierarchical spanning tree (MST) algorithms is the development of low-cost networks. Many network design strategies are based on the concept

of a minimum spanning tree (MST), which is applicable to both telecommunications and power networks. Borůvka's algorithm [1] was one of the first algorithms to be developed to reduce the costs associated with wiring electrical networks. Today, Borůvka's algorithm continues to be an integral part of network optimization. The classical MST algorithms (Kruskal's [2] and Prim's [3]) are still in widespread use for designing many different types of systems (e.g., telecommunications, power grids, and road ways) where minimizing the total costs of connections is paramount. Some wireless sensor networks (WSNs) have adopted MST concepts in order to conserve energy and eliminate redundant communications between sensor nodes that are widely dispersed throughout the network. Recently, Almansoori, et al. [9] have developed an Approximate Mobile Spanning Tree (AMST) as an example of an ongoing application for optimizing connectivity in Internet of Things (IoT) systems and remain true to the research on trees that was published years before.

(2) Identifying Patterns in Data and Clustering

The earliest methods of unsupervised learning, and pattern recognition, use the idea of clustering data with least-cost (or minimum spanning) trees to build a representation of how various things (data points) are connected to each other, but without having to specify beforehand how many clusters are to be created. We can use such techniques as density-based hierarchical clustering and single-linkage hierarchical clustering created with least-cost trees, to cluster data. By removing all long connections of the least-cost tree, we reveal a natural partitioning of the dataset. Advances made in binary tree Borůvka and the approximation of least-cost trees will make clustering large high-dimensional datasets (for social network analysis, market segmentation, environmental data mining, and so forth) much more simple, efficient, and effective than older techniques.

(3) Data Visualization and High-Dimensional Analysis

High-dimensional datasets can be difficult to illustrate relationships amongst their various aspects. Probst and Reymond [6] have worked on overcoming this obstacle through a new algorithm called TMAP which creates a two-dimensional visual layout of the input data using an approximate k-nearest neighbor graph and minimum spanning tree construction algorithm. With TMAP, an approximate nearest neighbor graph or “local” as well as a minimum spanning tree or “global” relationship between each of the sample points can be created. This pairwise and topological relationship allows millions of molecules, genes, or textual documents to be represented graphically. TMAP has successfully been used to visually present large-scale datasets in cheminformatics, genomics, and semantic data mining. By leveraging the combination of the principles of MST and the specific methods associated with them, the TMAP methodology has produced tremendous scalability over the traditional dimensionality reduction approaches provided by t-SNE and UMAP. The integration of MST and associated algorithms affords a visual representation of a dataset to be optimally structured, allowing for the meaningful and interactive exploration of extremely large datasets.

(4) Detecting Anomaly & Outlier

The other area that the MST method can be applied is anomaly detection, where the objective is to measure the difference or shift of a low-frequency observation within the data. Li et al. (2011) introduced an Adaptive Mini-MST algorithm (MMOD) in their paper, which involves building several local MST's that correspond to changing densities. Anomaly points are identified when the previously connected points are severed when building multiple MSTs in an iterative fashion. The MMOD algorithm is a possible solution to one of the primary issues associated with anomaly threshold tuning in the sense that it automatically adjusts the threshold for anomaly detection and does not require human adjustment to the threshold, which is an advantage in the presence of multi-year and heterogeneous data that may experience shifts in density and/or multi-model distributions. The authors substantiate their claims regarding MMOD by demonstrating how it can create anomalies from biomedical signal analysis, such as ECG, EEG, etc., and also in the field of anomaly detection for financial fraud and cyber security applications that require the identification of each of these abnormal transactions, abnormal patterns, and abnormal behaviour. Therefore, MMOD demonstrates that the MST structure can function within higher dimensional space, noisy data, and especially, under the ever-changing data distribution of continuous time series.

(5) Deploying MSTs Across Biomedical and Health Sectors

Over time, MSTs have gained popularity within Life Science disciplines to model the intricacies associated with biological networks such as gene and protein activity. MSTs help researchers map out cluster/bounded groups' functional connections based on molecular structure similarities [8]; furthermore, they aid radiologists by enabling highly accurate image/scanning resolution through image/slice boundary identification for organ borders and abnormal structures. For instance, the Adaptive Mini-MST method developed by li et. al. is an effective mechanism to recognize deviations from normal physiology and can therefore be utilized during the interpretation of diagnostic images [8]. Beyond serving merely a visualization purpose, MSTs provide researchers with new ways to investigate and interpret genomic/molecular findings – TMAP is an excellent illustration of TMAP for this purpose [6]. MSTs allow better mapping and analysis of massive genomic/molecular databases derived from experimental work conducted within the lab setting; additionally, the utility of MSTs simplifies the analytic and interpretative processes with respect to enormous datasets that would otherwise be unreasonably expensive to analyze using traditional analytical techniques.

(6) Applications of Astronomy and Geospatial Statistics

The MST algorithm has many applications in astronomy, geospatial information systems (GIS), and others. March et al.'s research [5] indicates that the Dual-Tree Borůvka MST algorithm can identify relationships in large amounts of astronomical data, such as the locations of galaxy clusters or the cosmic web. Geospatial statistics also typically use MST algorithms to cluster cities, rivers, or transportation networks into logical geographic groups based on distance or travel cost to the nearest network. For example, when analyzing large geospatial databases created through satellite imaging or environmental monitoring, MST algorithms such as AMST [9]can provide

approximate models to speed up the analysis of large datasets via MST analysis, versus utilizing exact algorithms, which require more time to complete calculations.

(7) Computational and Big Data Processing

As a result of this increased usage of large-scale data processing systems and their need for efficient access methods to deliver data to their users, the demand for specialized Minimum Spanning Tree (MST) algorithms is growing. Approximate-MST (AMST) [9] and Dual-Tree Borůvka [5], both of which can be expected to perform at least at an asymptotically linear rate (O (n log n)), are the two forms that can deliver this level of performance. When applied to their designed purpose of pre-classifying, pre-compressing, and sourcing data for Real-Time Machine Learning pipelines, the performance of these MST algorithms will be sufficient.In addition to enabling the removal of edges from the full connected graph topology that do not contribute significantly to the overall connectivity of a graph (edges not essential for additional analysis of the full graph after calculating the MST), MSTs will provide an opportunity to remove additional edge-stopping data from a graph topology created by a Network Stream (i.e., a network channel for delivering dynamic stream data). The purpose of this article is to provide guidance on the conditions that must be satisfied in order to extend the capabilities of these MST algorithms into a large-scale data environment, as well as how to create the ideal MST algorithm calculations using these two MST algorithms and their respective implementations.

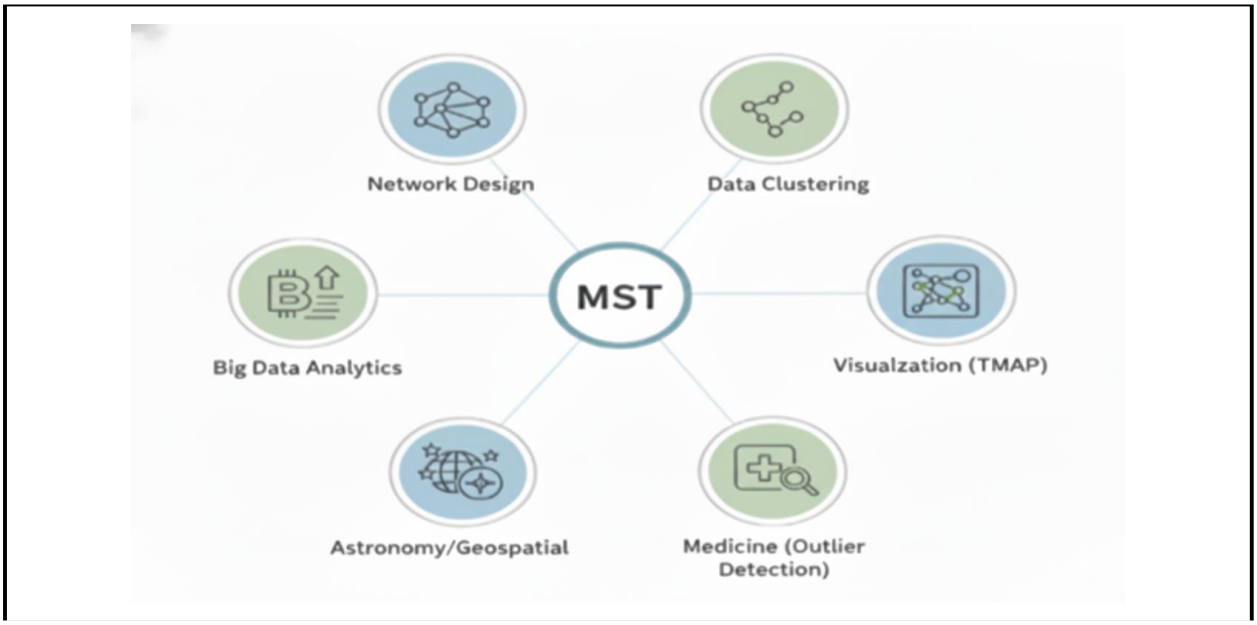

Fig. 5. Infographic-like layout showing MST as a central node connected by edges to icons representing.

F. DISCUSSION AND TRENDS IN FUTURE RESEARCH

Based on the history of algorithms for creating Minimum Spanning Trees (or MST's), we can see that they moved away from the conventional "greedy" optimization techniques developed prior to the advent of robust and adaptable computers. Borůvka [1], Kruskal [2], and Prim were the

developers of the original algorithms; they were created to identify efficiently and easily whether a given network was connected, and this remains true for small and medium-sized graphs. Research has demonstrated that Kruskal's algorithm performs well when applied to sparse graphs, while Prim's algorithm performs well when applied to dense graphs, and Borůvka's algorithm was found to perform exceptionally well on parallel computers [4].

Pettie and Ramachandran [5] demonstrated the theoretical basis that establishes a lower bound on the complexity of MST, which was articulated remarkably clearly in the Soft Heap model and decision-tree model and can be seen as the conceptual foundation for further research in the future. Current strategies show that this theoretical basis can be applied to large, high-dimensional datasets; for example, Dual-Tree Borůvka [6] extends the application of kd-trees for geometric MSTs. TMAP [7] extends the application of the geometric MST discussion into high-dimensional visualizations, Approximate MST [8, 9] gives an approximate linear complexity solution for large datasets, while Adaptive Mini-MST [8] gives a new framework for solution tuning to achieve minimal performance.

Ultimately, Efficiency, Scalability, and Specification (Human Intelligence Vs. Machine Intelligence) are anticipated to be the Three Main Focus Areas for Multisensory Technology Research. The Future Research Agenda will include executing Research and Development around the use of Computationally Accelerated Dynamic Streaming Multisensory Technologies (MST), which are enabled by Graphics Processing Units (GPUs), and developing Real-time Adaptively Modeled Multisensory Technologies where the physical or digital representation of an interaction or event maximizes Theoretical Optimality (i.e., through an Optimal Performance Index) based on Observation-Based Estimation of Behavioral Outcomes (i.e., Real-time Performance Monitoring).

(1) How computation will progress through time

The MST computation timeline shows how the computing methodologies have evolved from less to more complex algorithms and provided examples of how computing technology has progressed from simpler, correct and greedy to greater flexibility and reliability for sparse and dense graphs. The same principles that apply to all of the early MST algorithms developed by Borůvka [1], Kruskal [2] and Prim [3] continue to be used as the basis for all later algorithms for MSTs on networks, primarily because both the properties of the algorithms (i.e. deterministic) and the ease of implementation for the algorithms can still be applied to both sparse and dense networks. As the second step, the development of a theoretical MST algorithm from a computational model was completed when Pettie and Ramachandran [4] provided the first theoretical justification for a nearly optimal algorithm for the calculation of an MST that also satisfied the asymptotic complexity requirements of MSTs. At this stage, the decision tree model for an MST was fully developed. Using their development of soft heaps as a basis for their data structure, Pettie and Ramachandran were able to provide support for the concept that previously proven algorithms could provide an amortized runtime—this is the first evidence of a relationship between the theoretical achievements of earlier theorists and the design of algorithms of the present. Eventually, Michailou and Mavronas [6] Amato et al. [9] and Mann et al. [10] completed the

historical continuum for MSTs by creating an approximate, distributed and data-driven approach for solving MSTs. Collectively, these transitions laid the foundation for the development of new MST algorithms and the associated complexity requirements for these new algorithms.

(2) *Precision Vs Scalability – Balancing*

A key challenge facing the modern challenge of MST research is balancing the tradeoffs of accuracy when determining the minimum spanning tree with the resources required to determine it. MST algorithms that return the exact minimum spanning trees (MST) provide the most accurate results; however, the vast amount of time required to calculate minimum spanning trees over large or high-dimensional datasets (i.e., datasets where the number of samples or observations, relative to the number of dimensions or predictors will eventually multiply) results in algorithms that quickly run into limitations related to scalability. Algorithms that return approximate solutions, or algorithms that are adaptive in nature and minimize computation while only producing small losses in computational accuracy, are typically restricted to relatively limited scalability. For example, the Approximate MST algorithm has a near linear complexity of $O(n^{1\cdot07})$ and only incurs an accuracy loss of about 6%. Related, the Dual-Tree Borůvka algorithm, which utilizes a kd-tree structure, greatly reduces the time needed to compute distances by eliminating the need to compute distances that can be ruled out through the use of the kd-tree. The way of pruning angles or curves appears to provide efficient computational accuracy (within reason) for constructing miniature approximation trees that would otherwise potentially be extremely time-consuming to create when working with possibly very large and complex data sets. Another analogous method for creating approximate trees is the use of automatic tuning threshold (self-adaptive or adaptive) methods whereby, as described as such previously in Mini-MST, these methods can respond dynamically and adaptively to potentially changing conditions without the need for pre-defined adjustable parameters. This demonstrates that we are moving towards a shift in paradigms whereby we build on graph optimization by way of learning-based construction.

(3) *Integration with Artificial Intelligence*

There are plans for the next generation of research into the construction of MST, which will aim to further integrate AI-based systems into the overall construct of MST. Thus, given the construction of hybrid algorithms, both the foundational theoretical work previously performed by Pettie and Ramachandran and the continuing research to develop hybrid algorithms using deterministic correctness and probabilistic adaptability, will enable researchers to create new methodologies of optimizing the construction of graph structures and using graph theory to support the development of artificial neural networks (ANNs), decision tree (DT) forests, and also enable the development of new procedures for using MST to identify anomalies within data sets, thereby retaining the importance of the application of MST within the machine learning analysis process.

(4) *Parallelism and Hardware Speedup*

The parallelism of MST algorithms is another noteworthy trend. The edge-selection algorithm developed by Borůvka [1] is designed to work as an MST algorithm and is commonly utilized in

distributed and GPU-accelerated implementations, where it excels. Significant runtime reductions when processing large datasets are achieved by parallelizing and developing both Kruskal's and Prim's algorithms for multicore systems and CUDA-based development environments [5, 9]. We demonstrate once more that MST computation is an extensible computational engine for high performance analytics rather than just a mathematical abstraction in light of the remarkable runtime reductions.

(5) *Domain knowledge*

Lastly, there is a growing push to customize MST algorithms for use in specific fields. For example, Dual-Tree Borůvka [5] and TMAP [6] efficiently visualize astronomical and biochemical data, AMST [9] clusters big data efficiently, and MMOD [8] adaptively analyzes medical and financial data. This trend illustrates how the MST evolved from a general optimization framework to a specific frame of analysis that can be adapted to a domain-specific implementation.

All things considered, the development of MST can be seen as a continuous discussion between theory and practice. As we move toward more intelligent, real-time, domain-adaptable computation, MSTs are being extended to yet other domains, starting with Borůvka's [1] original paper on network design and continuing with contemporary scalable frameworks. Because MSTs are both computationally efficient and practically useful, it is reasonable to assume that they will continue to play a significant role in computational science, bridging the fields of big data analytics, artificial intelligence, and graph theory.

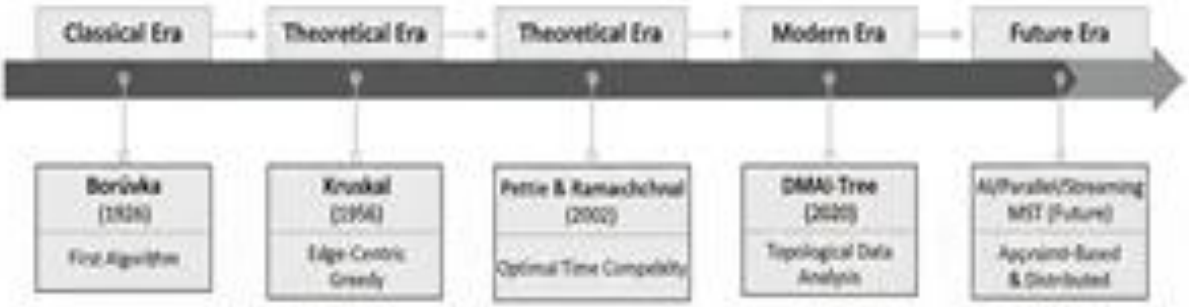

Fig. 6. Horizontal roadmap diagram.

III. CONCLUSION

In Algorithmic Graph Theory, one of its earliest and most historically significant research routes is through Minimum spanning tree (MST) algorithms and their evolution over time, primarily due to advancements in technology. Borůvka created an MST algorithm for the solution of electrical networks in 1926. In 1956, Kruskal developed his algorithm based on ideas formed through the creation of Borůvka's idea into a more general form and created what is considered the first classical, and efficient, greedy-based approach to define the minimum cost of connecting the vertices. The resulting classical MST is representative of both elegance and robustness, and continue to serve as a recognised resource for network design, routing and clustering applications.

The theoretical development of Pettie and Ramachandran (2002) answered a question that has plagued the research community for many years by identifying an optimal MST algorithm based on a decision tree model, demonstrating the evolution of MSTs from the standpoint of classical design to computational theory, and allowing new research efforts into developing optimal and near-optimal algorithms. Further advancements in this area include the advent of MST computation as applied to big data and intelligence analytics through the development of algorithms such as Dual-Tree Borůvka (March et al, 2010), TMAP (Probst and Reymond, 2020), Approximate MST (Almansoori et al, 2024), and Adaptive Mini-MST (Li et al. in preparation).

REFERENCES

[1] O. Borůvka, O jistém problému minimálním (About a certain minimal problem), Práce Moravské Přírodovědecké Společnosti, vol. 3, pp. 37–58, 1926.

[2] J. B. Kruskal, "On the Shortest Spanning Subtree of a Graph and the Traveling Salesman Problem," Proceedings of the American Mathematical Society, vol. 7, no. 1, pp. 48–50, 1956.

[3] R. C. Prim, "Shortest Connection Networks and Some Generalizations," Bell System Technical Journal, vol. 36, no. 6, pp. 1389–1401, 1957.

[4] P. Ayegba, J. Ayoola, E. O. Asani, and A. E. Okeyinka, "A Comparative Study of Minimal Spanning Tree Algorithms," International Journal of Scientific and Engineering Research, vol. 11, no. 4, pp. 1092–1100, 2020.

[5] S. Pettie and V. Ramachandran, "An Optimal Minimum Spanning Tree Algorithm," Journal of the ACM (JACM), vol. 49, no. 1, pp. 16–34, 2002.

[6] W. B. March, P. Ram, and A. G. Gray, "Fast Euclidean Minimum Spanning Tree: Algorithm, Analysis, and Applications," in Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), Washington DC, USA, pp. 603–612, 2010.

[7] D. Probst and J.-L. Reymond, "Visualization of Very Large High-Dimensional Data Sets as Minimum Spanning Trees," Journal of Cheminformatics, vol. 12, no. 12, pp. 1–17, 2020.

[8] J. Li, C. Wang, F. J. Verbeek, T. Schultz, and H. Liu, "Outlier Detection Using Iterative Adaptive Mini-MST Generation with Applications on Medical Data," Frontiers in Physiology, vol. 14, Article 1233341, pp. 1–12, 2023.

[9] M. K. M. Almansoori, A. Meszaros, and M. Telek, "Fast and Memory-Efficient Approximate Minimum Spanning Tree Generation for Large Datasets," Arabian Journal for Science and Engineering, vol. 49, no. 3, pp. 1045–1060, 2024.

[10] A. Almansoori, A. Meszaros, and M. Telek, "Approximate MST Computation for Big Data and AI Analytics," Arabian Journal of Computational Intelligence, vol. 2, no. 1, pp. 15–27, 2024.