Artificial Intelligence (AI)-Based Cross-Platform Node Name Mapping (Cmap-NN)

A Technical Paper prepared for SCTE by

Jordan Kupersmith
Consultant
Cox Communications
6205-B Peachtree Dunwoody Road NE, Atlanta, GA 30328
404-915-6955
Jordan.kupersmith@cox.com

Wei Cai
Lead Data Scientist
Cox Communications
6205-B Peachtree Dunwoody Road NE, Atlanta, GA 30328
404-414-7876
Wei.cai@cox.com

Jeonpaolo Barvez, Cox Communications

Pavan Chandrashekar, Cox Communications

Richard Brown, Cox Communications

© 2023, SCTE®, CableLabs® and, NCTA. All rights reserved.

Table of Contents

List of Figures

© 2023, SCTE® CableLabs® and NCTA. All rights reserved.

List of Tables

© 2023, SCTE® CableLabs® and NCTA. All rights reserved.

1. Introduction

The Strategic Network Planning and Automation team is the team at Cox Communications that is responsible for defining and applying strategic network growth policies, forecasting node level throughput, and creating a short to long-range plan for node actions From there, the plan is used by Cox's Outside Planning team, which plans anything downstream from the hub and Inside Planning team, which plans in the hub, to morph them into an operational plan. The plan drives millions of dollars in node actions annually for Cox. These node actions are vital for keeping the Cox customer connected and ensuring the quality of their connected experience. The node actions are also key for ensuring the continued growth and competitive viability for the company. Having the planning as accurate as possible so the right actions are being conducted at the right times and in the right places are necessary.

The team faces and has overcome many challenges and continues to work on the challenges they encounter daily. For example, the team has created highly predictive modeling using statistical and machine learning models to tune the model forecasts and provide greater accuracy over the years. Additionally, the team has created rules-based optimizations to create a more efficient planning process and limit the need for multiple touches on the node. This not only improves the customer experience, but it also saves money.

However, as the network grows more capable, it also grows more complex. With that, another set of challenges the team arises, which is the challenge of poor data quality. This is a growing problem as the complexity and size of the network continuously grows. With addresses mapped to the incorrect node, or node names not matching across databases the situation leads to errors in the field, and this can often lead to issues that impact customers.

As such, the team was challenged to couple its network expertise with their refined data science capabilities to define a machine learning algorithm that can not only identify mistakes in node attributes but predict the correct information. In this paper, we will dive into the technical aspects of the logic and the results of the proof of concept that one day will lead to greater accuracy in cross-system alignment, and practically eliminate errors, resulting in a more accurate planning process and avoid the costs associated with mistakes.

Cox relies on the outside plant (OSP) optical node name heavily as a major key in joining different systems together. In some instances, this can be the only key that may be used to join two or more systems together. As such, when there are discrepancies in one system for what the correct node name should be, this can cause a fallout in reporting or in some cases, customers receiving incorrect speed tiers or incorrect nodes being actioned for congestion relief and other upgrades. Three of the core systems Cox uses which has a significant influence on the node name are Geographic Name Information System. (GNIS), the collective Data Over Cable Service Interface Specification (DOCSIS) systems (CMTS and Cisco SmartPhy) and the Integrated Communications Operations Management System (ICOMS).

Cox uses a GNIS software system to maintain primarily outside plant network topology connectivity and assets. This system is critical to providing connectivity of customer to RF tap (and ultimately their optical node) as intended by the network design teams. The Hybrid Fiber-Coaxial (HFC) Design team is the originating stakeholders of which customers connect to which node. This name to topology association is ultimately passed to other systems such as ICOMS and inputted in the Cable Modem Termination System (CMTS). The topology information in GNIS can be associated with customers' billing information in ICOMS via their street address.

© 2023, SCTE® CableLabs® and NCTA. All rights reserved.

Commonly known as DOCSIS, this protocol is the international standard for IP traffic over coax. As mentioned in the context of this paper, it will collectively refer to the CMTS and SmartPhy data. The CMTS has a couple areas in the configuration where the node name may be written in the description field. Similarly, SmartPhy also allows for a "Remote Phy Device (RPD) Name" field which enables us to populate it with the Outside Plant (OSP) node name. The SmartPhy node name can be ensured it matches the node name in the service group description field by way of the chassis and Fiber Node ID field, which is a unique key for matching when combined. The customer identifiable information in the CMTS would be a list of customer premise equipment (CPE) MAC addresses reporting to each interface, which can be used to correlate to the MAC addresses in ICOMS and establish a physical location of each customer's device.

ICOMS is the third main source being leveraged. This is Cox's main billing system which contains certain customer identifiable data such as OSP node name, CPE, MAC address, and physical address. The MAC address can be mapped to a MAC address in the CMTS to match the node name per the CMTS via chassis and interface.

Unfortunately, data sources, especially large ones that are being constantly changed, are prone to discrepancies. While the interface to MAC address data in the CMTS is always correct, the device (node) name inputted in the CMTS/SmartPhy is made via manual entry. The node name in ICOMS is manual entry every time a new home becomes serviceable, or a node action causes a node name change. The GNIS OSP design data is all manual entry. Due to the ultimate human factor involved in node name entry for all core systems, this does pose some challenges to the machine learning factor (as it always has with the human factor).

Table 1 – Systems Overview

System	Node Name Entry Method	Level	Node originator
DOCSIS	Manual	Node	N
ICOMS	Manual	Customer	N
GNIS	Automatic	Node	Y

The current methodology for identifying discrepancies in the node name between systems is a dynamic reactive method. This is identified at two different granularities: one at the customer level and two, at the node level. The customer level tends to have the least impact traditionally of the two, with the primary impact of a single or staggered few customers within a node having the incorrect node name being subject to the possibility of services being sold that cannot be supported on their node. The node level granularity, which is when most of the customers within a node have the incorrect node name, can cause the incorrect CMTS interface to be associated to the OSP node name. This can result in node actions on a node that does not need one or higher/lower speed tiers being assigned to all customers in a node which cannot support those tiers of service. There are currently no teams nor anyone working or discovering these discrepancies as part of their official duties. Employees from various departments will discover these node name discrepancies accidently as they are impacted by them and submit them for a fix.

To address this problem, a Machine Learning based node name determination solution is proposed in this paper. If two systems have a street address, a fuzzy-string matching algorithm can establish linkage. For systems that can only be linked with location coordinates a, distance matrix and GMM clustering model are built.

© 2023, SCTE® CableLabs® and NCTA. All rights reserved.

The rest of this paper is organized as follows. In Section 2, we introduce the theoretical description of the chosen string similarity algorithm with the focus on the fuzzy string matching. The other algorithm – Haversine Distance used in this paper is also discussed. We further introduce theoretical principles behind the Gaussian Mixture Models (hereafter GMM) Clustering Algorithm, which is concluded with a summary of a related work. In Section 3 we describe the dataset used for our studies, together with the data sanitization used prior to the Extract Transform and Load (ETL) Scoring, Haversine Distance analysis and GMM clustering models, and present methodologies used in this paper along with our proposed hybrid scoring algorithm. In Section 4 we focus on presenting results and improvements accomplished by using the proposed model algorithm in mapping nodes across systems. Further research and next steps are summarized in Section 5.

2. Background and Literature Review

The way that street addresses are structured can vary, their components can be unstable, and they are prone to different human entry errors (Coetzee and Rademeyer 2009). When comparing them between two systems they are not always exact strings. While postal addresses are structured information since it contains easily identifiable and delimited fields such as city, postal code and street number, they can be written several ways through the use of abbreviations, synonymous and contractions, resulting in the possibility of non-exact strings for the same postal address between two systems (Perez et al 2008). Due to these constraints, postal addresses between two systems cannot be joined directly nor is it possible to generate a list from one to be used to filter the other. For this reason, Natural Language Processing (NLP) techniques must be used.

The fundamental aspect of the possibility of being able to match addresses is the algorithm used for measuring distance and dissimilarity between two postal addresses. There are several algorithms to choose from and they have a variety of techniques. There are algorithms that measure the changes required to convert an expression to the other that compare their textual structure, vocal structure, or a combination of both methods. Examples of textual methods are Guth, Hamming, Levenshtein, and Damerau–Levenstein, all of which measure the edit distance between two expressions counting the inserts, deletes, and substitutions required to transform them. Vocal methods such as Soundex, Metaphone and Phonex leverage the phonetics of words to analyze their similarity. Finally, there are methods like Jaccard and Cosine, which analyze a combination of string and vocal structures to provide probabilities to represent the similarity of the expressions (Rezayan et al 2018).

2.1. Fuzzy String Matching and Related Work

Fuzzy string matching is a technique used to find approximate matches between two strings. Algorithms may be divided into two categories due to the feature they measure:

- similarity algorithms: the match is found if S(X, Y) ≥ t_S,
- dissimilarity algorithms: the match is found if D(X, Y) ≥ t_D,

where $t_{S/D}$ is a string similarity/similarity threshold, S(X, Y) and D(X, Y) are the similarity and dissimilarity functions, X and Y are the two strings in question. When two datasets share only a single imperfect identifier, this is sometimes called the *fuzzy string matching problem* (Filipov and Varbanov 2019; Hall and Dowling 1980).

Previous research has shown that the best algorithms for detecting one road name written in two different ways are Mongo Elka soundex Jaro, Jaro Winkler, and Levenshtein in that the specified order when, for

© 2023, SCTE® CableLabs® and NCTA. All rights reserved.

example, comparing “Avenue J.F. KENNEDY” with “Avenue J-F KENNEDY”. However, when comparing two strings that are not the same such as “RUEDESARDENNES” and “RUEDESJARDINS”, all previously mentioned algorithms except for Levenshtein produce high similarity scores, meaning they have high false positive rates (Charif et al). In all classification problems, one must always choose whether it is more important to minimize false negatives or false positives, and the preference varies by the context. In the context of this paper, false positives are more detrimental than false negatives because failing to identify a match is not as determinantal as having false matches due to the quantity of matches in the system. If one match fails, there are others to stand in its place. For this reason, Levenshtein distance is the best suited algorithm for these purposes.

Furthermore, speed is just as important as accuracy. Fuzzy Matching on strings requires calculating the cartesian product of both datasets before calculating the similarity score between strings. The Cartesian Product is defined as the set consisting of all ordered pairs. For example, if $A = \{x, y\}$ and $B = \{3, 6, 9\}$, then $A \times B = \{(x, 3), (x, 6), (x, 9), (y, 3), (y, 6), (y, 9)\}$ (Encyclopedia Britannica). Given that there are millions of customers, it is important to be able to implement the algorithm quickly. Levenshtein Distance is not the fastest algorithm, but it is also not the slowest according to previous research (Christen 2006). However, Python, the programming language which is used for the majority of the study, has a library called FuzzyWuzzy that easily implements the Levenshtein distance in parallel, fully being able to harness the processing power of the 40 CPU system that it runs on. For the above-mentioned reasons, we selected Levenhstein Distance for string matching.

Levenshtein Distance is a mathematical measure used to calculate the difference or similarity between two strings of characters. It counts the minimum number of operations needed to transform one string into another, where each operation can be either an insertion (adding a character), a deletion (removing a character), or a substitution (changing a character). In short, the Levenshtein Distance tells you how many changes you would need to make to turn one word or phrase into another, giving you a measure of their overall similarity or dissimilarity. (Levenshtein 1966). For example, if the source string is “maple” and the target string is also “maple” then the distance is 0 because no transformations are needed, and the strings are identical to each other. On the other hand, if we compare Poplar ST and Popular AVE there is a Levenshtein Distance of 4 because to transform the source string into the target first we must add a “u”, then substitute an “s” with an “a”, then substitute a “t” with a “v” and finally add an “e”. Hence, the higher the Levenshtein Distance, the more different the strings are. The final distance is then displayed as 100-total distance count (Chen et al 2014).

$$\operatorname{lev}_{a,b}(i,j) = \begin{cases} \max(i,j) & \text{if } \min(i,j) = 0, \\ \min \begin{cases} \operatorname{lev}_{a,b}(i-1,j) + 1 \\ \operatorname{lev}_{a,b}(i,j-1) + 1 \\ \operatorname{lev}_{a,b}(i-1,j-1) + 1_{(a_i \neq b_j)} \end{cases} & \text{otherwise.} \end{cases}$$

Figure 1 - Levenshtein Distance Algorithm

© 2023, SCTE® CableLabs® and NCTA. All rights reserved.

2.2. Haversine Distance

The Haversine Distance is a well-established method to compute the distance between two points represented as latitude-longitude coordinates. This formula was first presented by James Inman in 1835 (Inman, 1849), and serves as a foundational technique in numerous geographical information systems.

$$haversin(d/r) = haversin(\emptyset_{(}2-)\emptyset_1) + cos(\emptyset_1) + cos(\emptyset_{(}2))haversin(\lambda_{(}2-)\lambda_1)$$

Figure 2 - Haversine Distance Equation

Haversine Distance has been used to determine the shortest distance between two points on the surface of a sphere, given their longitude and latitude. This is especially useful in the context of finding location similarity, as it provides a method to calculate the distance between two geographic coordinates. Historically, the formula has been used in navigation and cartography, allowing sailors to determine the shortest path between two points on a globe before the advent of electronic navigational tools. There have been multiple research papers that discuss or implement the Haversine formula. For instance, research into efficient algorithms for geospatial data, location recommendation systems, and spatial network analyses often leverage the Haversine formula for distance computations.

2.3. Gaussian Mixture Model (GMM) Clustering

GMM is a mixed density model with a basic Gaussian distribution, which is equivalent to the weighted average of multiple Gaussian probability density functions. Each Gaussian density function is called a component, which represents a cluster. In the following description of GMM algorithm, α_ι denotes the weight of each Gaussian distribution in GMM, and μ_i, α_i^2 denote the mean and the covariance matrix respectively.

$$\mathrm{P(t)} = f(x) = \sum_{i=1}^{N} \alpha_i N(t; \mu_i, \alpha_i^2)(\alpha_i \geq 0, \Sigma_{i=1}^{N} \alpha_i = 1)$$

Figure 3 - Gaussian Mixture Model Algorithm

The robustness of GMM stands out when compared to traditional clustering techniques. While K-means clustering and Hierarchical clustering segment the data based on distance or hierarchical structures, and Density-based clustering (DBSCAN) operates on the density distribution of data points, the GMM lies in its ability to approximate continuous distributions with high accuracy, even when the component count escalates. This precision becomes particularly advantageous when addressing the spatial distribution of intricate datasets, like the ones examined in our study.

Our research sought to utilize the GMM's capabilities in generating cluster fingerprints that aptly represent the spatial distribution of our cross-platform nodes. Another intrinsic advantage of the GMM is its soft clustering feature. Unlike hard clustering methods that definitively assign data points to a specific cluster, GMM computes the probabilities of a data point belonging to different clusters. This probabilistic assignment becomes imperative when handling data points that are ambiguous and could potentially fit into multiple clusters. By leveraging these probabilities, our study derives the maximum likelihood for each data

© 2023, SCTE® CableLabs® and NCTA. All rights reserved.

point. This probabilistic output was subsequently transformed into a scoring component in our proposed algorithm, further aiding in the detailed interpretation and categorization of our data.

3. Methodology

The burgeoning field of computational techniques provides insights into intricate problems, one such being the identification of identical nodes and gauging the accuracy of node names. Guided by existing literature, particularly on fuzzy string matching, Haversine distance, and GMM clustering in spatial data, we create an innovative intersection. This nexus aims at harnessing computational prowess to mirror human cognitive thinking processes that intuitively recognize node similarities and discrepancies. There are numerous advantages, especially considering machine learning (ML) paradigms, renowned for swiftly discerning potential matches and mismatches-surpassing human speeds and possibly, accuracy.

As foundational steps, our methodology encapsulates both data sampling and subsequent data processing. Before diving deep into algorithms and ensemble learning, it becomes imperative to understand the data's structure, its origins, and the preliminary operations performed to make it amenable to advanced analytical processes.

Due to the volume of data and its processing time involved, only one geographic area was selected for the purpose of the study. Over 7,350 of nodes' data were selected as the dataset used in our analysis. Arizona was chosen because it has the greatest number of nodes that have gone digital, and thus resulted in a lot more node name changes.

In our methodology, we first procured low-level geographical data, specifically latitude and longitude coordinates, from each respective system under study. These granular data points were subsequently aggregated, culminating in a node-level dataset. This consolidated dataset then becomes pivotal as it serves as the input for our subsequent processes.

© 2023, SCTE® CableLabs® and NCTA. All rights reserved.

3.1. Levenshtein Distance Matching and ETL Scoring

In this dataset, we follow the below data model to collect data from each system:

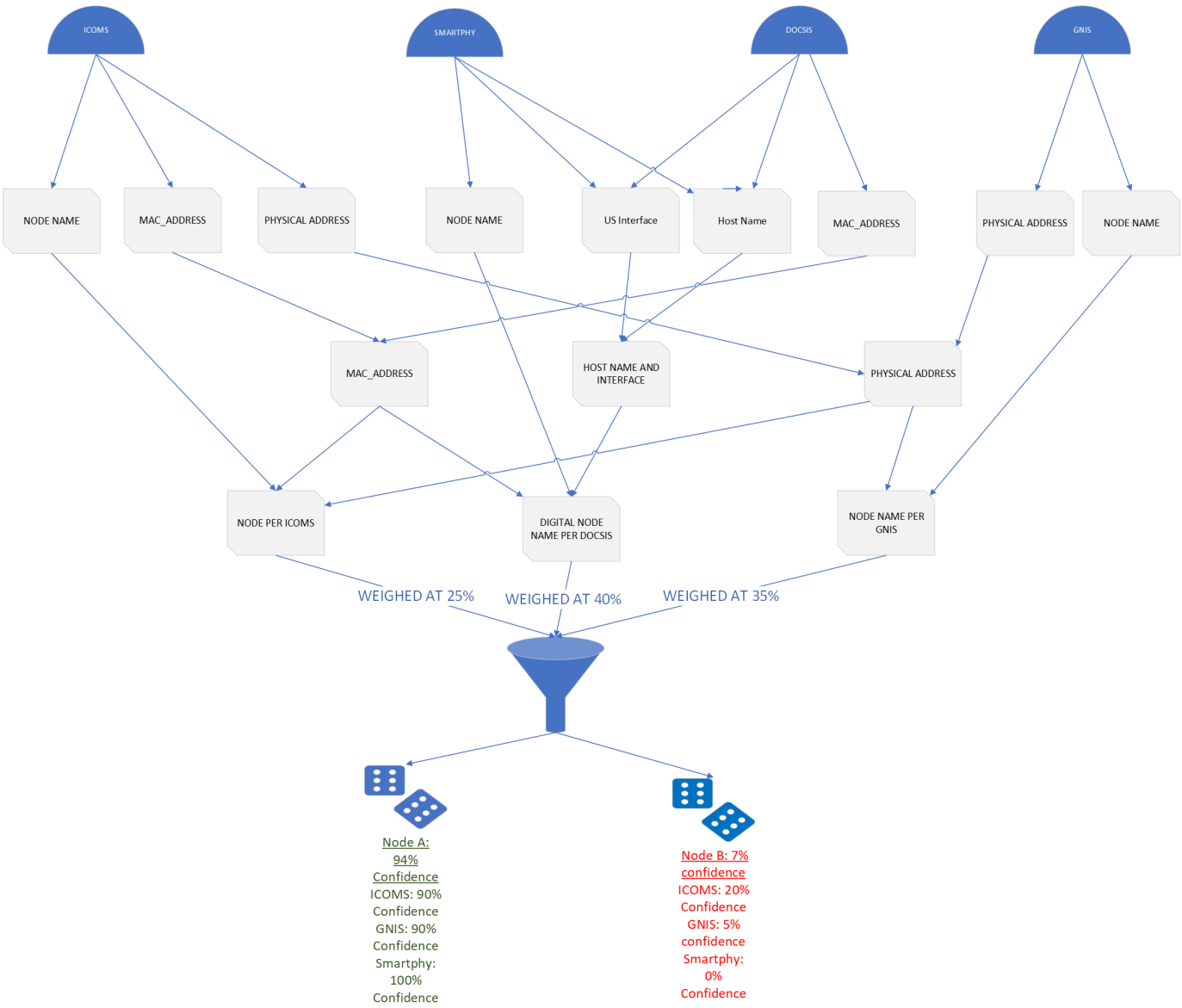

Figure 4 - Diagram of the Matching System

© 2023, SCTE® CableLabs® and NCTA. All rights reserved.

From the above data model diagram, we can see that while we cannot use node name to establishing links between the systems in all cases, there are ways to link them together. The main challenge between linking them together is that the links between each system are not always the same variable, and the format of these variables is not always the same due to different standards of writing street addresses between systems. In other words, an exact string join is not possible. Furthermore, these systems rely on many records of MAC addresses to be linked together, and while joining them would be straightforward if they were in the same database, these systems are not connected in any way, therefore it requires creative computational techniques to be able to link them.

The system begins by examining a node in SmartPhy. Each node has a host name and interface name associated with it, which is how this data source relates to DOCSIS. The tables are systems that are on the same database with slightly different text format, so after some string manipulation they can be easily joined together. Additionally, unlike subsequent steps, this join is done at the node level, meaning that it does not present any challenges of big data.

Nonetheless, all subsequent steps are done at the MAC address and street address level, meaning that millions of records need to be reconciled, and it is not possible to match these using simple database technology. Matching these records is only possible through programmatic techniques designed to meet the challenges posed both by big data and the fact that the data lies on disparate data systems.

Due to limits of RAM, this processing must be done by looping through nodes and writing output to the disk to minimize consumption of RAM at a given time. A list is generated of all of the nodes in the SmartPhy/DOCSIS dataset and that list is used to iterate through the dataset by node. In the loop, a temporary dataset is created that contains all MAC addresses per DOCSIS node, which is an average 1,500 in each node.

The MAC addresses from DOCSIS are then looked up in the data-frame for ICOMS data, resulting in the first cross-system node mapping of the program. This is because while we have a node name in DOCSIS and we have a nodename in ICOMS, we cannot trust that a given node name points to the correct combination of device and customer. In other words, let's say that Customer A has device 1. We have node XY1 associated with Customer A and we have node LM2 associated with device 1. While we can prove that device 1 belongs to customer A because they share a MAC address, we have no idea which node name is right. For this reason, we must line up customers and devices to know when node names coincide and when they do not. This process matches DOCSIS with ICOMS.

The next step takes our combined DOCSIS/ICOMS dataset and matches it to GNIS based on street address.

Given there are millions of customers, checking every single address in one system against every single address in another system is computationally too expensive. If Cox had only 1 million customers, this would be 1 trillion calculations. If each calculation took 1 second, this would take 31,709 years. While this could be parallelized, with 30 cores it only speeds it up to 1,000 years. If we were to quadruple the speed, then it would be reduced to 250 years. Even if it is done for just one node it would take 42 days per node at 1 second per address without multiprocessing. It is clear that NLP alone cannot solve this problem.

For this reason, it is necessary to pre filter the data before submitting it for NLP matching. For a given node in SmartPhy, all the zip codes in that node are extracted in the ICOMS dataset after passing it through the filter of MAC addresses in DOCSIS and these zip codes are then used as a filter to reduce the My World dataset to contain only entries that have the same zip codes which are present in ICOMS for this node. Subsequently, all the street numbers present in ICOMS are extracted and used to filter the My World dataset

© 2023, SCTE® CableLabs® and NCTA. All rights reserved.

to only contain records where the street number is present. By reducing the My World dataset to only a combination of zip codes and street numbers which are present in the ICOMS dataset, it can be filtered, which is possible because numeric data can easily be matched precisely, greatly reducing the number of records that need to be fed into NLP.

Given the large amount of data, this operation is run in parallel, using one less core than the maximum available number of cores and the amount of time this takes varies widely by node. There are nodes that can finish in 25 seconds and other nodes that take longer than 20 minutes. The result is a matrix of all address combinations from ICOMS, with the address candidates from My World and their score. This matrix is combined back onto the original ICOMS and My World datasets and they are merged based on the address string they contributed to the matrix which results in a combined dataset of all systems. In this dataset, there is a variable for node name for all 3 systems, the shared mac address between DOCSIS and ICOMS, the shared street address between ICOMS and My World as well as special attributes which are unique to only one system such as latitude and longitude which is only found in My World.

The matched data is then loaded into a loop where for each time a SmartPhy node appears per combination of MAC address and street address, each node name according to GNIS is counted. For example, if there are 100 combinations of MAC address and street address for node A and GNIS agrees that the node name is node A 70 times, but it says it is node B 40 times, the program returns a dictionary that attributes this count of 70/40 to the SmartPhy node in question. Nodes that make up less than 2% of the GNIS dictionary are removed to minimize noise. These counts are then translated into percentages and interpreted as the percentage that GNIS agrees with. In this example, the count of 70 and 40 can be translated to a percentage of 63.6% and 36.3%, respectively. These percentages can be interpreted as “GNIS agrees that node A is really called node A 63.6% of the time while GNIS believes 36.3% of the time that what SmartPhy calls node A should actually be called node B.” This same logic is repeated for each combination of MAC address and street address for all entries of SmartPhy’s node A for ICOMS to compare the ICOMS system to SmartPhy as well.

To examine a real world example, the third node in Table2 is 8ANPL as recorded by the SmartPhy system. For every MAC address that belongs to this node, we see that My World/GNIS calls this node by the same name only 32 times as opposed to calling it 8ANPJ 47 times. When also factoring in the lowest occurring node name, 8ANPK, this node is present in My World 106 times. The total percentage of confidence that My World gives to this node is 32/106=30.2%. This contrasts with the last row in the example that contains 8BSV2, where My World agrees 100% of the time resulting in a confidence score of 100% as well as the first row that contains 8EDG2 where My World does not cast even one vote of confidence in the node name, resulting in a confidence score of 40%, the minimum confidence for the SmartPhy system.

© 2023, SCTE® CableLabs® and NCTA. All rights reserved.

Table 2 – Smartphy by GNIS Count

SmartPhy	my_world	my_world_vote_for_SmartPhy	my_world_vote_pct_for_SmartPhy
8EDG2	{'01326': 18, '8EDG1': 328}	0	0.0%
8DWE1	{'00407': 252}	0	0.0%
8ANPL	{'8ANPJ': 47, '8ANPK': 27, '8ANPL': 32}	32	30.2%
8AEHB	{'8AEHB': 217}	217	100.0%
8BSV2	{'8BSV2': 298}	298	100.0%

All these percentages are recorded in a dictionary and stored in a dataframe to be associated with the SmartPhy nodes. For the example of 8ANPL, GNIS agrees with the node name in 30.2% of occurrences as shown in Table 2. Furthermore, ICOMS agrees with the node name in 29.5% of the occurrences. Since we are considering the information that SmartPhy presents, it automatically casts a vote for itself of 100%. These confidence votes are used in the weighted mean formula (Figure 5) with the weights of 40% for DOCSIS, 25% for ICOMS and 35% for GNIS as established in the introduction section of the paper. This results in an overall confidence of 57.9% that when SmartPhy claims it is the node name that it is indeed true.

$$\bar{X}_w = \frac{\Sigma w_i x_i}{\Sigma w_i}$$

Figure 5 - Weighted Mean Formula

While envisioning the framework of the machine learning model, a "weight" was decided to be assigned to each of the three main sources. A weight of 40% was assigned to DOCSIS, 35% to GNIS, and lastly 25% to ICOMS. A weight system was decided due to certain sources observed over the years that tend to have less discrepancies than others. The weights are also helpful being different between each source as this would especially help in situations where all three systems display a different node name. The node name in the DOCSIS-based systems (the CMTS and SmartPhy) tend to be the most accurate so we assigned it a weight of 40%. The GNIS system is meant to be the origination point of where all node names are created initially so one would assume that would have the greatest weight. However, the GNIS node name flowing through downstream databases first is not always the case as there can be a long delay before a design post a node action is pushed to a production state. This will result in all topology assets still indicating the old and now incorrect node name. The ICOMS system was given the least weight as it has been observed to have the most discrepancies over the years.

The same process of getting counts by systems is repeated two more times, counting the number of times a node name appears in GNIS and SmartPhy for each node name as recorded in the ICOMS system and counting the number of times a node name appears in ICOMS and SmartPhy for each node name according to GNIS. When examining the same node name from the perspective of the ICOMS and GNIS systems in our example of 8ANPL, it can result in difference confidence scores. For every instance where ICOMS claims a node to be named 8ANPL, GNIS agrees 100% of the time. However, SmartPhy only

© 2023, SCTE® CableLabs® and NCTA. All rights reserved.

agrees 33.3% of the time. Given that ICOMS votes for itself at 100% and its weight is 25%, the confidence score is 73.3% which can be interpreted as being a 73.3% confidence that when ICOMS claims a node is named 8ANPL, that it is true. When examining the same node from the perspective of GNIS/My World, ICOMS agrees with it 96.9% of the time and SmartPhy agrees with it 33.3% of the time. Given that GNIS votes for itself at 100% with a weight of 35%, it also results in a confidence score of 73.3%. This can be interpreted as being a 73.3% confidence that when GNIS claims a node is named 8ANPL, that it is true.

For each node in SmartPhy, the name of the node that occurs most frequently according to the systems is also extracted. For the example of 8ANPL, with 47 votes the most frequently occurring node according to My World is 8ANPJ. In this example, ICOMS provides the exact same confidence votes as My World, but this is not always the case. Since the most frequent node name is not the same across all systems, the confidence of this alternative node is name is also calculated, which is 44% for both My World and ICOMS and 0% for SmartPhy. The result is 26.7% confidence for this alternative node name. While these calculations are not conclusive for this node, this is not always the case. In the example of 8EDG2 from Table 2, this node name is not found in My World at all while My World has a 94.8% confidence that the actual node name is 8EDG1. ICOMS exhibits a confidence of 95.9% for this node name as well. The overall weighted mean for this alternative node name is 57.1%. The reason it is not higher is because SmartPhy, the most highly weighted system, attributes a vote of 0%. While not an extremely high confidence, it is still higher than the confidence of 40% for the default node name. 40% represents the minimum confidence for any node name according to SmartPhy and is assigned to node when the other systems have 0% agreement of the node name. If the weight of SmartPhy were to decrease, this minimum confidence would decrease, and the alternative node confidence rate would increase. The alternative node name calculations are repeated two more times as well, using ICOMS and GNIS as the reference point for nodes and using the other two remaining systems to cross check its validity.

The ETL scoring approach, while effective in many aspects, faces a notable limitation in its inability to capitalize on matching based on geographic latitude and longitude data. These coordinates hold significant predictive value in determining the accurate designation of node names, given the pivotal role that geography plays in distinguishing between nodes, irrespective of nomenclature. Neglecting these coordinates in the analysis could be seen as an omission, which calls for additional methodologies to be included in the algorithm.

3.2. Haversine Distance

In the distance analysis process, we need to consider two issues: how to choose the cutoff distance, and how to calculate the similarity score.

3.2.1. Choice of the Cutoff Distance

Selecting the cutoff distance to determine location similarity depends on the specific context and requirements of your application. In our schema, the cutoff distance defines the maximum distance between two locations for them to be considered similar. The choice of the cutoff distance will influence the granularity of the clustering or grouping of cross-system nodes. A cut off distance that is too high may lead to false positive cases where two different nodes are identified as the same node. Meanwhile, a cut off distance that is too low may lead to false negative cases where same nodes in two different systems mistakenly are classified as different nodes. In our scheme implementation, the network experts' insights were used to determine an appropriate cutoff distance range.

© 2023, SCTE® CableLabs® and NCTA. All rights reserved.

However, selecting the cutoff distance is not always a straightforward process, and requires some trial and error. Hence, cross-validation was used to evaluate the performance of different cutoff distances on a hold-out dataset to achieve the best performance. We start with a small cutoff distance, and gradually increase it to observe how the similarity patterns change. This incremental approach allows us to explore different levels of granularity in the node mapping. The final cut-off value based on multiple trials was set to be 1.25 km.

3.2.2. Calculation of the Node Similarity Score

The node distance similarity score is based on the ratio of the Haversine Distance and cutoff distance. Namely, we express the percentage of similarity of two nodes as follows:

$$Similarity\ Score\ (\%) = abs\ (1 - \frac{\text{Haversine Distance}}{\text{Cutoff Distance}}) * 100\%$$

Figure 6 – Node Similarity Score

Here, the Haversine Distance, derived from the equation in Figure 6 specifies the distance in kilometers between nodes from distinct systems. Meanwhile, the Cutoff Distance represents the chosen threshold to adjudicate node similarity, as informed by our iterative assessment.

In the context of node mapping, a match pertains to nodes whose coordinates are either identical or proximal within the cutoff distance range. Our operational premise posits that nodes with minimal distances between them are more likely to be identical, and vice versa.

3.3. GMM Clustering

In the realm of data clustering, the Gaussian Mixture Model (GMM) has emerged as a pivotal tool, particularly when dealing with multifaceted distributions in datasets. The GMM method functions by approximating the underlying data distribution using a combination of multiple Gaussian distributions, typically identified via the Expectation-Maximization (EM) algorithms. Our research has incorporated this GMM clustering methodology, emphasizing its role in ascertaining node similarity scores, a key metric for analyzing and understanding node-related traffic pattern and strategic node action plan.

By harnessing the GMM's capabilities and incorporating a robust range of statistical measures, we believe our method offers a holistic view of node similarity, thereby enhancing the precision and relevance of our results in node mapping.

3.4. Ensemble of the Methodologies

As previously discussed, the ETL scoring method has limitations in that latitude and longitude are not considered by it. The Haversine and GMM methodologies also have limitations, particularly considering that the boundaries of nodes are often fluid, encompassing varying distances from a few yards to several miles. This complexity is compounded by the distribution of individual customers around node centers in diverse directions, leading to instances where a customer might be closer to a node center that is not their actual assigned node.

Such intricacies render geographic coordinates insufficient in unequivocally identifying a node; however, they serve as valuable tools in the elimination of potential node candidates. For instance, in Figure 7, a hypothetical example illustrates that geography alone might not definitively determine whether a depicted customer belongs to Node B or C, even though they are in closer proximity to the center of Node B, they

© 2023, SCTE® CableLabs® and NCTA. All rights reserved.

belong to Node C. Despite its inability to precisely pinpoint node designation, geographic data does aid in excluding certain nodes from consideration. For example, in Figure 7 it can be inferred that the customer does not belong to node A.

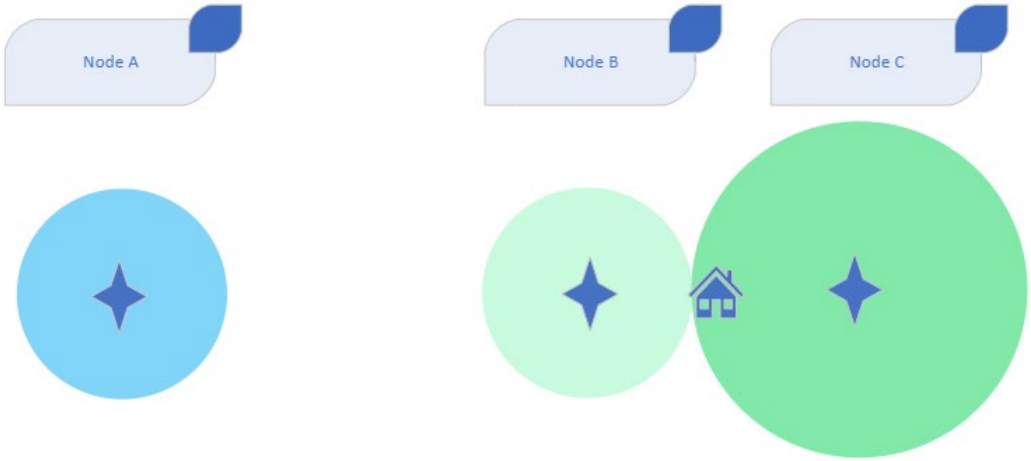

Figure 7 - Node Geographic Distribution

Due to the natural constraints of the cross geographic analysis, it is used as a supplemental analysis to either affirm or refute the conclusions from the ETL method, which was found by engineers to not always point to the correct node names. The resulting uncertainty is quantified through the confidence probabilities associated with these approaches, and when combined with the ETL method, the overall confidence score is modified. The final score is determined by taking a weighted average of all three scores, with the ETL method having a weight of 85%, the Haversine Method having a weight of 10% and the GMM method having a weight of 5%.

4. Results

In this section, we present a comprehensive demonstration of the enhanced efficacy achieved over existing best practices, substantiating our claims through discernible results and discoveries. Our objective here is to establish the superior capabilities of the ensemble learning algorithm in generating better-matched node names within the context of applied research.

4.1. Levenshtein ETL Scoring Method

Of the 5,582 Arizona SmartPhy nodes that were considered, 4,388 produced results. 3,190 nodes were found to be in complete agreement between all three systems. However, while highly correlated, agreement between systems is not synonymous with 100% confidence because agreement is defined as GNIS and ICOMS pointing to the same node name as the node name that has the most occurrences between all systems but there can still be other node names with nearly as high frequencies suggested by GNIS and ICOMS. 1,152 nodes of the SmartPhy nodes have 100% confidence and another 1,664 nodes have 90-99.9% confidence, all of which classify as agreeing. Of the nodes with under 90% confidence, only 373 of them classify as being in complete agreement.

© 2023, SCTE® CableLabs® and NCTA. All rights reserved.

Overall, regardless of whether a node is flagged as having complete agreement or not, there are 1,152 nodes with 100% confidence and 89 nodes with 40% confidence, which is the minimum confidence and is defined as not finding any support for the node name in the other two systems. The complete confidence distribution is displayed in Figure 8. Of these nodes, there are 54 nodes that have an alternative node suggested which have a higher confidence score than the SmartPhy node name and there are a total of 92 nodes that alternative nodes with higher confidence when combining more than one alternative node. The number of nodes that have a higher scoring alternative node name is limited given that they have a minimum confidence score of 40% due to the weights given to SmartPhy.

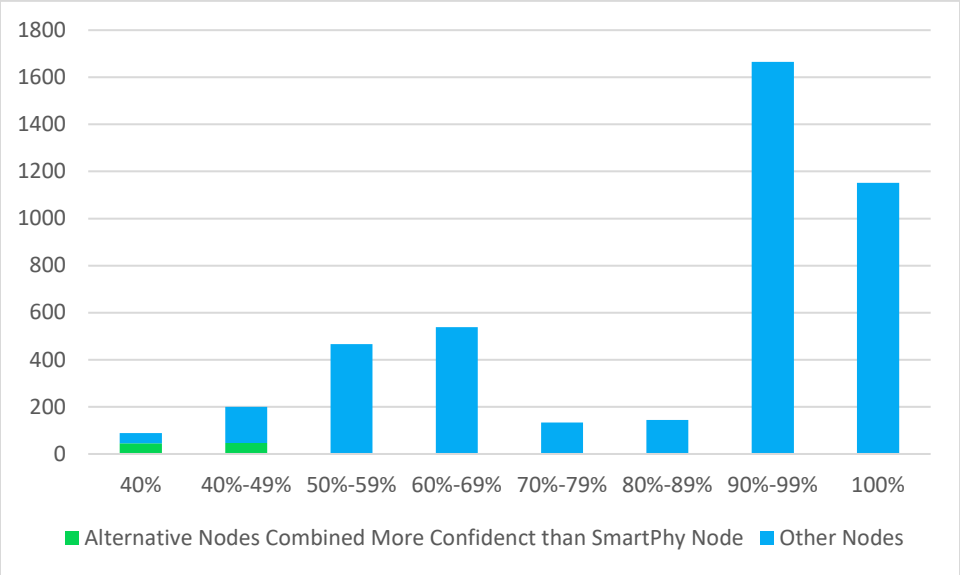

Figure 8 - Confidence Distribution of Results for SmartPhy Nodes

There are 6,483 nodes according to GNIS, which is slightly more than SmartPhy. With 377 nodes having the minimum confidence score of 35%, it is likely that there is a bit more noise in the node names in My World since there are only 89 nodes with the minimum score in DOCSIS. Of the nodes with a minimum score, 68% of them have alternative node names with a higher confidence score, suggesting the system is effective in suggesting what is likely the true node name when the default node name is in doubt. Looking towards the nodes with high confidence, there is a drop in the number of nodes that have 100% confidence from SmartPhy. In SmartPhy, these nodes are 26% of all nodes while in GNIS they are only 6% of all nodes. However, the 90-99% bin in GNIS represents 39% of all nodes, which is slightly more than the 37% of nodes in this bin in SmartPhy. Irrespective of confidence scores, 3,238 nodes, or 50% of nodes have agreement with the two other systems, meaning that the other systems cast a majority vote for the GNIS node name even if they have other alternative node names to suggest.

© 2023, SCTE® CableLabs® and NCTA. All rights reserved.

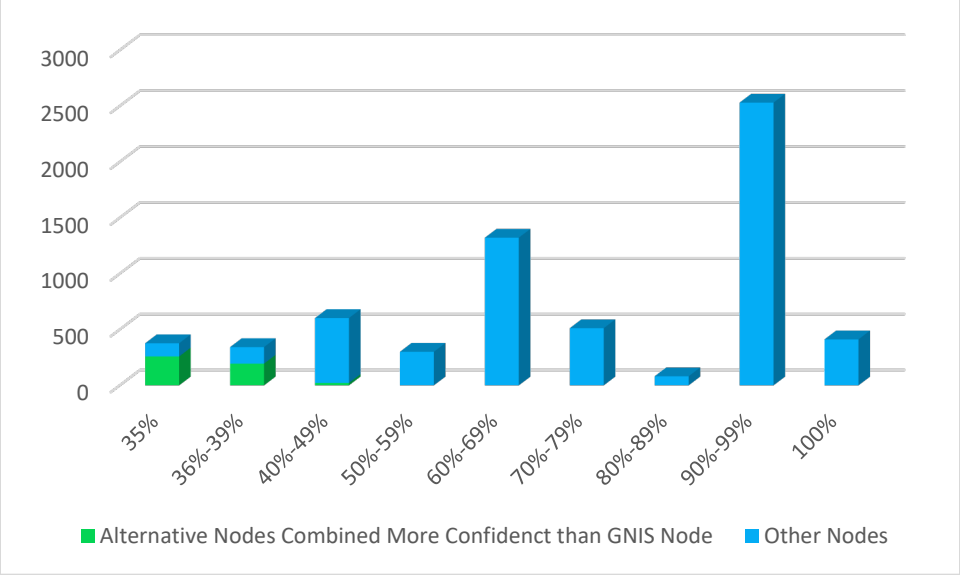

Figure 9 - Confidence Distibution of Results for GNIS Nodes

There are 10,292 nodes according to ICOMS, almost double the nodes of SmartPhy. Nonetheless, of these 10,292 nodes, 3,673 are likely noise as represented by having a percentage of 25%, which is the minimum for ICOMS. However, there are 1,037 nodes with a confidence of 25% or greater than the total amount of nodes in SmartPhy, suggesting that there are several nodes that the other systems are not capturing such as analog nodes which would not exist in SmartPhy or GNIS due to a node action. Furthermore, of 3,673 nodes with 25% confidence, the program suggests alternative node names with higher than 25% confidence in 3,091 of the nodes with 25% confidence. When examining all ICOMS nodes irrespective of their confidence score, there are 1,487 nodes whose alternative names have a confidence of 50% or greater, which is strong evidence that the system works. On the other side of the spectrum, there are 3,222 nodes where all node names are in complete agreement.

The complete confidence breakdown is displayed in Figure 10, which shows strong concentrations at the extremes of having complete confidence that the ICOMS node name is the correct node name as represented by 100% confidence as well as a confidence score of 25%, the minimum confidence, which represents 60% of all ICOMS nodes. There are not too many nodes at other confidence ranges except for a spike in the 60-69% confidence range. Of these 2,189 nodes with 60%-69% confidence, 1,874 of them have a very high confidence from GNIS and a very low confidence from SmartPhy. This may be indicative of a construction/splicing issue where the trunk and feeder cables were connected to the

© 2023, SCTE® CableLabs® and NCTA. All rights reserved.

incorrect side of a 2x2 node.

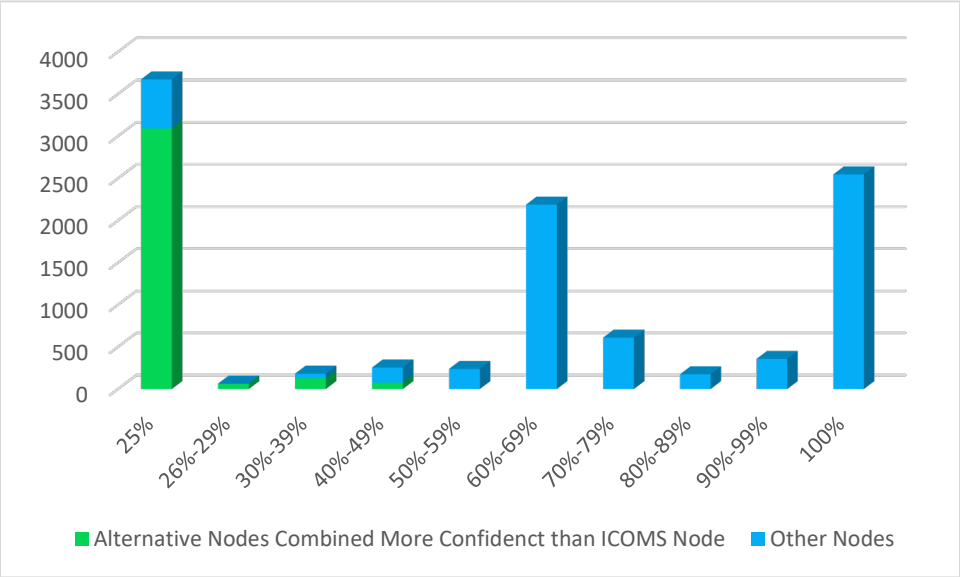

Figure 10 - Confidence Distibution of Results for ICOMS Nodes

Consistent with the weight selection, DOCSIS has the greatest node confidence followed by GNIS and finally ICOMS with 64.2%, 54.4% and 28.3% node confidence of 90% or greater in these systems respectively. DOCIS, GNIS and ICOMS have 6.6%, 20.4% and 40.4% node confidence of under 50%, respectively as well, showing consistency of the relative confidence of systems on both the high end and low end of confidence. While these confidence score differences are influenced by the weights, the gaps exceed the differences of weights between the systems which are 40% for DOCIS, 35% for GNIS and 25% for ICOMS, showing consistency with the weights and affirm the engineers' anecdotal observations about the difference of confidence between the systems.

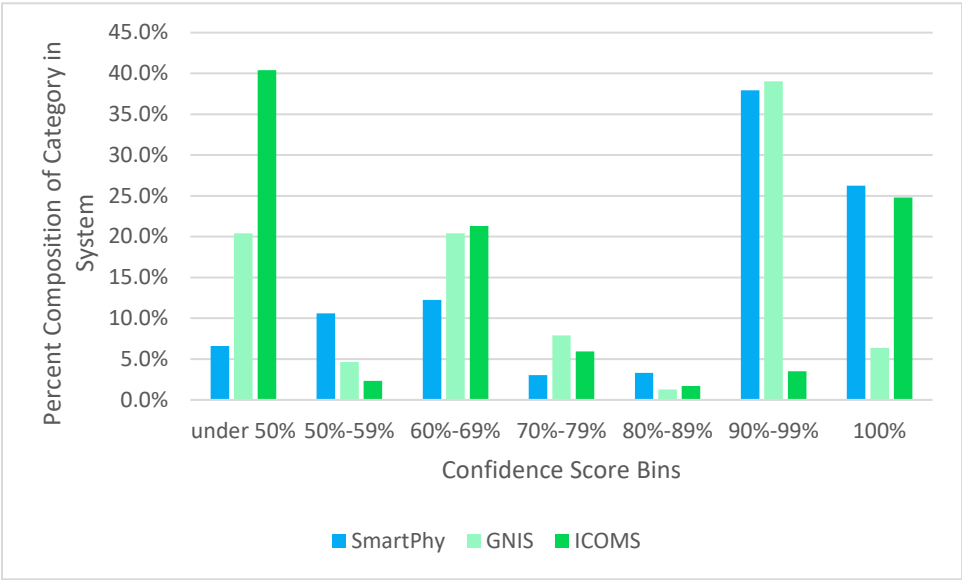

Figure 11 - Confidence Distibution Comparison Between Systems

© 2023, SCTE® CableLabs® and NCTA. All rights reserved.

4.2. Haversine and GMM Method

As discussed in the methodology section 3.4, a scenario may emerge where each standalone method fails to provide engineers with sufficient information to definitively determine the valid node name across platforms, particularly in cases where a node bears multiple unmatched names across platforms. In response, we propose a hybrid machine learning approach to enhance confidence in the node name mapping process. To validate this proposed learning methodology, we incorporate distance matrix scores and GMM (Gaussian Mixture Model) probabilities. These scores are weighted using subject expert defined parameters across three systems, yielding distinct scoring sets for each node: the ETL/Levenshtein FuzzyString Score, Haversine Distance score and the GMM clustering score. With these scores and corresponding confidence datasets, we execute matching at the node level and tabulate the counts and percentages of node names exhibiting high similarity scores (≥ 70 out of 100) in each method. This alignment is illustrated through a Venn diagram in Figure 12, where individual methods are denoted by separate circles of distinct colors, facilitating clear differentiation. Sky-blue signifies Levenshtein FuzzyString, violet purple denotes Haversine Distance, and a slightly darker blue represents GMM Clustering.

In Table 3, we provide real life examples of nodes that are in Venn Diagrams for further clarification about what the Venn Diagrams represent.

Table 3 – Sample Nodes For Venn Diagram Clasification

Node	ETL/Fuzzy Score	Haversine Score	GMM Cluster Score
8ADBA	79.4%	88.9%	96.7%
8BXH1	100.0%	97.9%	66.7%
8AXR2	84.9%	66.5%	64.3%
8DTS1	99.4%	78.3%	91.8%
8VGR1	98.7%	90.3%	89.6%

Node 8ADBA is in the purple region of Diagram A, the dark blue region of Diagram B and the intersection region of Diagram C. Node 8BXH1 is in the intersection of A, the teal of B and the purple of C. 8AXR2 is in the teal of A, the teal of B and is not in C. 8DTS1 is in the purple of A, the blue (intersection) of B and the blue of C. Finally, 8VGR1 is in the intersection of all graphs.

In Diagram (A), out of 12,333 node names with high scores, the Levenshtein FuzzyString Method has 9,896 with high scores, the Haversine method has 6,863 with high scores and 4,426 (36%) nodes are in agreement by having high confidence scores in both methods. Similarly, in Diagram (B), out of 9,946 node names highly scored by Levenshtein FuzzyString and 5,565 with high GMM Clustering scores, both methods have high confidence in the node name in 3,672 (31%) of nodes. Diagram (C) shows that out of the 6,863 nodes with high confidence in Haversine Distance and 5,565 nodes with high scores in GMM Clustering, both methods give high confidence in the node name in 3,415 (38%) nodes.

© 2023, SCTE® CableLabs® and NCTA. All rights reserved.

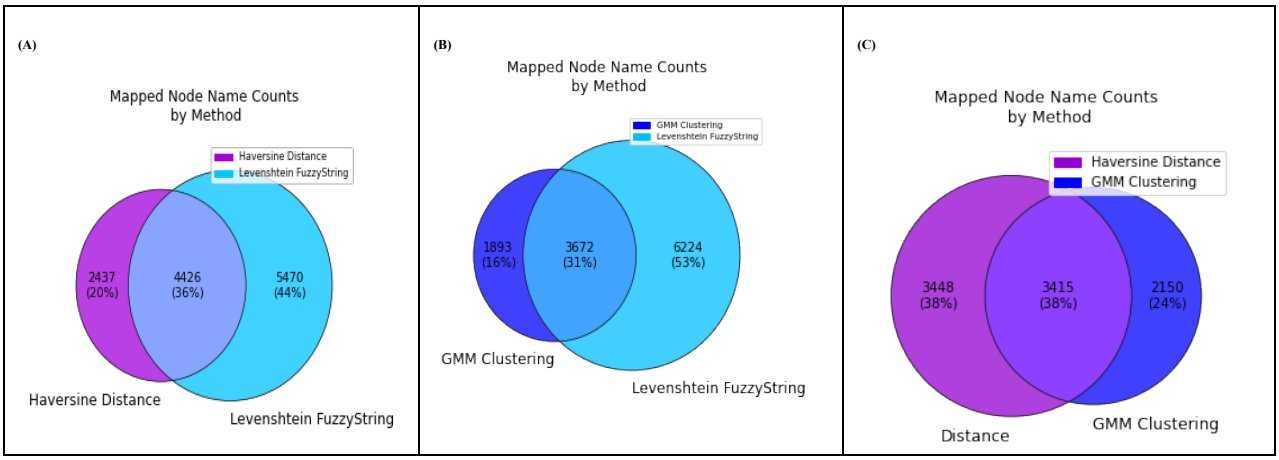

Figure 12 - Mapped Node Name Counts by Levenshtein FuzzyString, Haversine Distance and GMM Clustering

In our analysis, the overlapping region delineated by both GMM clustering and Haversine Distance is notably smaller than anticipated. We attribute this to two reasons:

(1) Amount of nodes with high scores: The count of nodes highly ranked by each method is relatively limited; the Haversine Distance method identified less than 7,000 nodes, while GMM clustering accounted for under 6,000. Nonetheless, having nearly 3,500 nodes recognized by both methods is still a significant overlap.

(2) Differences in Input Data: While both methods utilize the same raw data, there are certain disparities in the data inputs fed into each algorithm. For instance, consider the node labeled 8ARUC, which has 24 pairs of coordinates associated with it in the dataset. We engineered variables for the GMM model by calculating descriptive statistics including the mean, minimum, median, 95th percentile, and maximum of these 24 pairs, as detailed in Table 4.

Table 4 – Sample Node-Level Coordinates (eg Longitude and Latitude)

Node Name	Longitude Mean	Longitude Max	Longitude P50	Longitude P95	Longitude Min	Latitude Mean	Latitude Max	Latitude P50	Latitude P95
8ARUC	-111.927998	-111.926366	-111.927659	-111.926478	-111.932305	33.494212	33.494631	33.494228	33.494589

Only the median longitude and latitude (i.e., lon_p50 and lat_p50) were utilized to determine distance in the Haversine method. In contrast, GMM clustering employed the full array of these descriptive statistics. This discrepancy might illuminate why the overlapping region is smaller than initially projected.

4.3. Ensemble Method

While these individual methodologies demonstrate substantial alignment in node mapping, they also exhibit varying degrees of uncertainty when employed individually. To harness the strengths of each method, we propose an ensemble approach that amalgamates the three methods into a meta-learner. This ensemble

© 2023, SCTE® CableLabs® and NCTA. All rights reserved.

methodology seeks to yield a definitive consensus, and upon executing the same scoring and evaluation on equivalent datasets, we observe analogous performance results, as depicted in Figure 13.

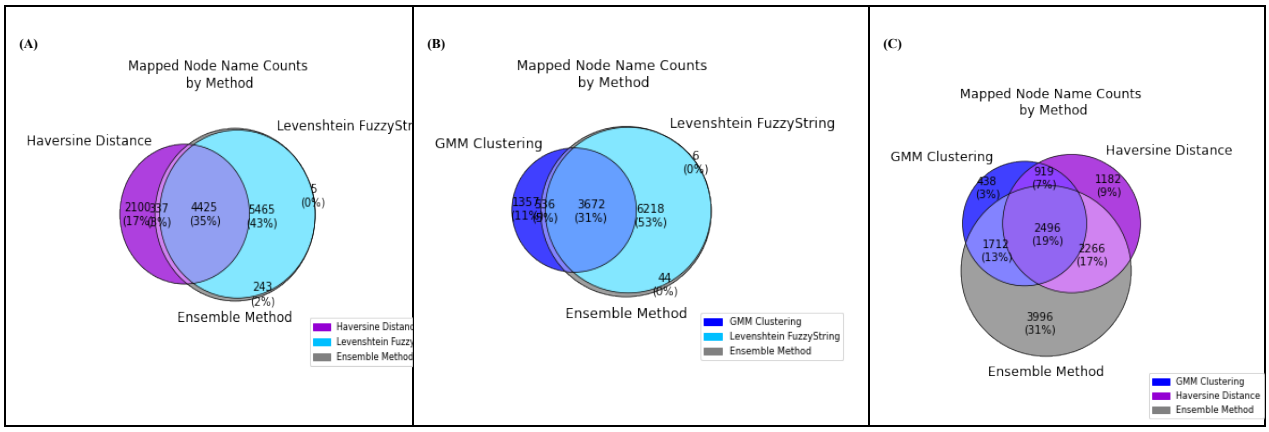

Figure 13 - Mapped Node Name Counts by Levenshtein FuzzyString, Haversine Distance, GMM Clustering, Ensemble Method

To aid in the understanding of the results, we have compiled a summary of the number of nodes captured by each method, accompanied by their respective percentages, in Table 5. It must be noted that in each Venn Diagram, it is comparing two methods with the combined method, meaning that one method is excluded in each Venn Diagram. For example, the gray area of Venn Diagram C shows that 31% of nodes belong to the ensemble method only category. However, these nodes are matches solely based on the Levenshtein Method which is not displayed in Venn Diagram C.

Table 5 – Ven Diagram Explanation

Method and Regions (Node Counts and Percentage)	(A)	Method and Regions (Node Counts and Percentage)	(B)	Method and Regions (Node Counts and Percentage)	(C)
Methods Used	(1) Haversine Distance; (2) Levenshtein FuzzyString, (3) Ensemble Method	Methods Used	(2) Levenshtein FuzzyString, (3) Ensemble Method; (4) GMM Clustering;	Methods Used	(1) Haversine Distance, (3) Ensemble Method; (4) GMM Clustering;
2,100 (17%)	Nodes with high similarity score unique to (1)	1,357 (11%)	Nodes with high similarity score unique to (4)	438 (3%)	Nodes with high similarity score unique to (4)
337 (3%)	Shared nodes with high similarity score in (1) and (3)	536 (5%)	Shared nodes with high similarity score in (3) and (4)	1,712 (13%)	Shared nodes with high similarity score in (3) and (4)

© 2023, SCTE® CableLabs® and NCTA. All rights reserved.

4,425 (35%)	Nodes with high similarity score that coexist in (1), (2) and (3)	3,672 (31%)	Nodes with high similarity score that coexist in (1), (2) and (3)	2,496 (19%)	Nodes with high similarity score that coexist in (1), (2) and (3)
				919 (7%)	Shared nodes with high similarity score in (1) and (4)
5,465 (43%)	Nodes with high similarity score in both (2) and (3)	6,218 (53%)	Nodes with high similarity score in both (2) and (4)	2,266 (17%)	Nodes with high similarity score in both (1) and (3)
				1,182 (9%)	Nodes with high similarity score unique to (1)
243 (2%)	Nodes with high similarity score unique to (3)	44 (0%)	Nodes with high similarity score unique to (3)	3,996 (31%)	Nodes with high similarity score unique to (3)

In Figure 13, a notable trend emerges: a reduction of 337 nodes that originally resided within the disagreement region between the Haversine Distance and FuzzString methods has been effectually shifted to the overlapping region between Haversine Distance method and our proposed Ensemble method. Although this yields a modest increase of just over 300 nodes in the shared overlapping region, the impact of this enhancement becomes more pronounced when considering the smaller population of nodes endowed with high-confidence scores in the Distance method.

Transitioning to Figure X1 diagram (B), a notable pattern emerges as 536 nodes now occupy the coveted category of targeted matched nodes within the ensemble framework. This outcome stands as a testament to the robustness and precision of the proposed approach in comparison with Fuzzy String Method.

Finally, the culmination of GMM clustering, Haversine Distance, and the Ensemble method offers intriguing insights. Upon closer scrutiny, a combined total of 5,457 nodes could be considered as nodes with high similarity score from three methods. Among these, 2,496 nodes are shared across three methods, 1,712 nodes are common between GMM and the Ensemble method, and 1,266 nodes coexist between Haversine Distance and Ensemble method. This shift exemplifies the capacity of the Ensemble method to bridge disparities and align node mappings in a more efficient manner.

5. Conclusion, Limitations and Further Research

5.1. Conclusion

With the objective of achieving automated node determination across disparate systems, we introduce a methodological framework underpinned by fuzzy postal address matching, location coordinate similarity, and unsupervised geographical clustering. This paradigm leverages an array of heterogeneous data sources,

© 2023, SCTE® CableLabs® and NCTA. All rights reserved.

encompassing mac addresses, home addresses, and longitudinal-latitude pairs, to meticulously delineate the identification of target matching nodes. Concomitantly, the methodology amalgamates ETL processes and an unsupervised machine learning model, thereby facilitating the cohesive evaluation of node similarity across heterogeneous platforms.

Through meticulously conducted experimental endeavors, the empirical findings notably underscore the appreciable amelioration in the mitigation of node misclassification discernible across distinct systems. It is very consequential in its implications and our proposed methodology demonstrates a performance trajectory that outpaces individual techniques when assessed within the context of cross-system node mapping.

The results of this paper serve as a proof of concept to demonstrate the potential of a Cross-Node Mapping System if resources are allocated to implementing it into production. By using Artificial Intelligence and modelling, we can continuously make improvements to the model and have the model self-learn to the point we can have node topology almost entirely AI-model driven and seamlessly incorporated with other tools and sources we have at our disposal once the system enters production. This will provide great benefits from a full automation and cost savings perspective.

5.2. Limitations

It is possible to improve the coverage of the system. While SmartPhy is regarded as the most reliable data source and is used as system that generates the list of nodes to consider for the system, it has a notable limitation in that only digital nodes are in it, meaning that for the purposes of this study analog nodes are excluded. It is estimated that there are 7,773 total nodes in Arizona but SmartPhy only has access to 5,872 nodes, meaning that there are approximately 1,901 analog nodes in Arizona that were excluded from this research. It is possible that these missing nodes could have associations with some of the 3,091 ICOMS nodes that appeared to be noise.

The lack of analog nodes is not the only cause of missing nodes in the study. Due to processing errors in the matching process loop, 4,388 out of 5,582 nodes were able to be considered. The code should be examined further to determine if it is capable of functioning on a portion of the 1,194 nodes that were not able to be processed.

One possible solution that can address missing nodes is to adjust the starting point of the intake process in the system. Currently, the proposed system takes the nodes that are in SmartPhy and loops through them to match to the other systems. As the ICOMS system is currently the system with the highest discrepancies, one possible adjustment is to loop through host name or interface name and feed those parameters into SmartPhy instead of simply the ICOMS or SmartPhy node name, which could be faster and capture more data. In Arizona, there are 142 CMTSs with each having 50-100 service grouping of line interfaces. Another potential approach is to start the system by looping through GNIS nodes generated directly by the design team when a node action occurs which could increase accuracy and trigger the system at the most critical times. Looping through the ICOMS nodes is another potential solution which could also provide greater coverage of nodes but could potentially be slower. Combined and shared-interface 2x2 nodes will pose a unique challenge at the moment when applying these nodes in the model as we currently cannot tell which node a modem is polling from. However, as modems on RPD nodes can be polled to know which US and DS port they are reporting back to even in the case of combined 2x2s, this will be something to refine the model logic and explore soon.

Finally, we found that latitude and longitude in the ICOMS database was not always accurate. To improve the Haversine Distance and GMM results it is necessary to improve the accuracy of these coordinates.

© 2023, SCTE® CableLabs® and NCTA. All rights reserved.

This can be achieved through expanding the use of the google maps API. However, it costs $500 per each block of 100,000 requests.

5.3. Next Steps

To put the system into production, it is necessary to automate the data extraction process, which is currently done manually by querying the Oracle database where the data is housed, creating Comma-separated values files, reducing their size via bash and reading them into pandas. A possible solution is to write a series of more restrictive queries that pull data by market and create jobs to refresh the source files on a scheduled basis.

Another challenge that must be addressed is to speed up the ETL matching process, which currently takes 50 hours for Arizona alone. To scale up to the whole country, it could take one to two weeks of processing time under the current framework, which is not sustainable and means by the time the ETL process is done it could be outdated. Methods to improve the speed include a more efficient implementation of multiprocessing and further development of the fuzzy matching method, which is the subprocess with the longest compute time. Several possibilities for improvement include the exploration of more pre-filtering, a more efficient use of multiprocessing, the exploration of faster fuzzy matching algorithms and the exploration of different implementations within Python of Levenshtein distance. Studies have shown that the Jaro algorithm is 70% faster (Christen 2006) which could reduce the processing time from 50 to 15 hours. Furthermore, the program to create the confidence score takes 22 minutes in Arizona and should also take less time prior to scaling up.

Furthermore, it is possible to increase the efficacy of the Geographical Gaussian Mixture Model (GMM) spatial clustering. Particularly when applied in isolation, this mechanism evinces a diminished efficacy in scenarios involving nodes proximate to each other. A possible enhancement for future research is to construct a successive clustering scheme based on the GMM cluster results to further enhance node matching precision.

To be useful to engineers it is also necessary to establish a user interface that they can interact with to see the confidence of node names and alternative node name suggestions. A simple web application could be the first step, followed by integration into their existing systems and creating alerts to provide more easily available analysis and to allow for taking proactive measures.

Another enhancement is to introduce apartments into the string matching which is necessary for accuracy in urban areas as some apartment buildings can share the same address but have multiple nodes. Nonetheless, this has its challenges because My World has its apartments as ranges. For example, an address in My World could be 172 S Main ST 253-268 Phoenix, AZ. This single record represents all apartments in that building from 253 to 268. On the other hand, ICOM has a separate record for each of those unit numbers. The data must be adjusted, and additional preprocessing is required to be able to introduce apartment numbers into fuzzy matching.

After the successful implementation of the cross-node mapping system for informational purposes, there can be potential new opportunities to enhance customer service. The GNIS platform has a web interface called My World which has geocoding of data but is limited to data in the GNIS database. By matching to ICOMS, additional information can be made available such as if a customer is active or not, which can help improve customer service.

Another potential functionality of the system is identifying ICOMS nodes that have few customers associated with them for a given DOCSIS node which can be indicative of people who moved to another

© 2023, SCTE® CableLabs® and NCTA. All rights reserved.

jurisdiction and took their router with them. This system could aid in automatically updating their node information in ICOMS which would also improve customer service.

Abbreviations

CMTS	Cable Modem Termination System
CPE	Customer Premise Equipment
DBSCAN	Density-based clustering
DOCSIS	Data Over Cable Service Interface Specification
EM	Expectation-Maximization
ETL	Extract Transform and Load
GMM	Gaussian Mixture Model
GNIS	Geographic Name Information System.
HFC	Hybrid Fiber-Coaxial
ICOMS	Integrated Communications Operations Management System
ML	Machine Learning
NLP	Natural Language Processing
OSP	Outside Plant
RPD	Remote Phy Device
SmartPhy	Cisco SmartPhy platform for configuring RPD nodes

Bibliography & References

"Operations on Sets." *Encyclopædia Britannica*, www.britannica.com/science/set-theory/Operations-on-sets#ref387948. Accessed 2 Aug. 2023.

Charif O, Schneider M. "How to Geolocalize an Administrative File Basing on its Addresses Data? A Proposal to Achieve Better Geocoding" OMRANI H. CEPS/INSTEAD, DIFFERDANGE, LUXENBOURG

Chen, Yoke Yie, et al. "Email Hoax Detection System Using Levenshtein Distance Method." *Journal of Computers*, vol. 9, no. 2, 2014.

Christen, Peter. "A Comparison of Personal Name Matching: Techniques and Practical Issues." *Sixth IEEE International Conference on Data Mining - Workshops (ICDMW'06)*, 2006, https://doi.org/10.1109/icdmw.2006.2.

Coetzee, S., & Rademeyer, M. "Testing the spatial adjacency match of the Intiendo address matching tool for geocoding of addresses with misleading suburb or place name." Paper presented at the 24th International Cartographic Conference, Santiago, Chile, November 15–21 2009.

© 2023, SCTE® CableLabs® and NCTA. All rights reserved.

Filipov, L., and Z. Varbanov. 2019. “On Fuzzy Matching of Strings.” Serdica Journal of Computing 13(1–2):71–80.

Hall, P. A. V., and G. R. Dowling. 1980. “Approximate String Matching.” ACM Computing Surveys (CSUR) 12(4):381–402.

Levenshtein V. “Binary Codes Capable of Correcting Deletions, Insertions, and Reversals. Soviet Physics Doklady10 (8); Feb 1966.p. 707–710.

Rezayan, Hani, et al. “Quality Evaluation of Postal Address Datasets Measuring Their Autocorrelation.” *GeoJournal*, vol. 84, no. 6, 2018, pp. 1617–1625, https://doi.org/10.1007/s10708-018-9940-x.

© 2023, SCTE® CableLabs® and NCTA. All rights reserved.