April 2022

SMART GROUPING – GOING BEYOND DOMAIN SIMILARITIES IN PEER RECOGNITION BY LEVERAGING COMPLEX HIERARCHIES USING MACHINE LEARNING

Doosan Jung

Qihong Shao

Belanna Zhou

Jonathan Arnowitz

Gurvinder Singh

See next page for additional authors

Inventor(s)
Doosan Jung, Qihong Shao, Belanna Zhou, Jonathan Arnowitz, Gurvinder Singh, and Kiran Yedavalli

SMART GROUPING – GOING BEYOND DOMAIN SIMILARITIES IN PEER RECOGNITION BY LEVERAGING COMPLEX HIERARCHIES USING MACHINE LEARNING

AUTHORS:
Doosan Jung
Qihong Shao
Belanna Zhou
Jonathan Arnowitz
Gurvinder Singh
Kiran Yedavalli

ABSTRACT

Techniques are presented herein that support an intelligent system for grouping customers by leveraging exactly those things that make it impossible for a person to perform effective clustering. That is, the presented techniques support clustering based on a customer's vastly complicated network profiles. Aspects of the presented techniques encompass a smart customer grouping framework with highly accurate deep learning modeling, utilize domain-specific machine learning (ML) to unravel the nonlinear latent representations with a deep autoencoder, provide a flexible feature weighting capability to focus on certain features based on customer personas and business, and support a highly usable system for sales and marketing professionals. Use of the presented techniques allows for the grouping of network customers, considering their complex hierarchical structure, using ML.

DETAILED DESCRIPTION

Confucius' famous quote – "To know what you know and what you do not know, that is true knowledge" – is now the bane of our existence. To earn a competitive advantage, understanding what one does not know is more and more essential. Techniques are presented herein that support a way for sales and marketing professionals to learn "what they do not know" in support of their pursuing new and unprecedented sales efforts to aid underperforming customers.

One of the most valuable activities for sales and marketing professionals concerns finding potential customers for an upsell opportunity, especially customers who are

unknown or whose need is unaware. To address that activity, aspects of the techniques presented herein support a three part solution. First, the professionals need to understand their customer baselines and benchmarks for operational excellence. Second, they need to know who the customers are that fall short of those measurements. Third, they need to understand customer analogs, identify the lower performing customers, and help those customers achieve optimal performance (e.g., a networking sales professional may need to help customers renew their licenses, upgrade their devices, add services, improve their architecture, etc.).

According to aspects of the techniques presented herein, to understand customers' needs, and to identify a potential upsell opportunity, there are two steps involved as shown in Figure 1, below.

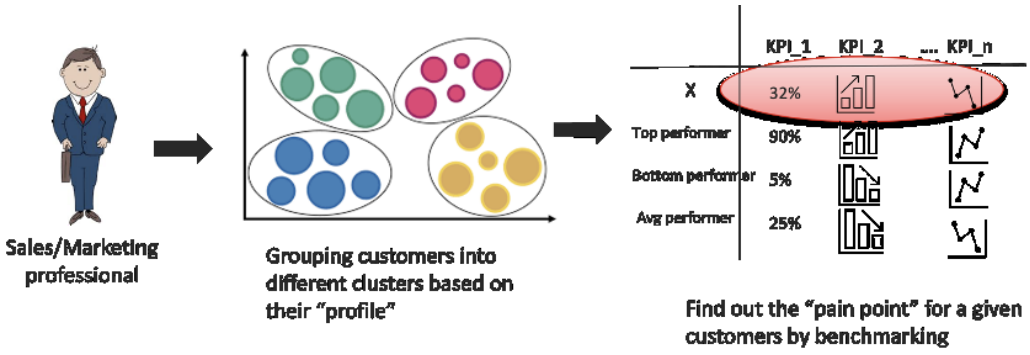

Figure 1: Smart Grouping and Customer Needs Analysis

Referring to Figure 1, above, a first step encompasses the grouping of customers based on a collection of objective key characteristics, along with their range of values, in which truly similar customers may be grouped. A second step encompasses, for each group, determining the performance matrix to identify who the low performers are and where they are lagging behind (e.g., comparing their performance with all of their peers and determining whether and where the customer needs help and providing actionable insights.)

The creation of the clusters as described above is a very challenging and time-consuming process. Each customer has many different devices and every device has very different situations (with regard to, for example, business goals, networking aims, security

concerns, end of life issues, bug fixes, etc.). Figures 2a and 2b, below, depicts elements of the circumstances that were described above.

Figure 2a Hierarchy Structure

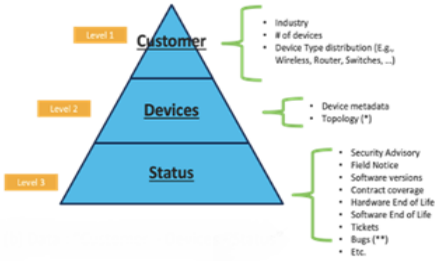

Figure 2b: Exemplary Data

Figure 2a, above, depicts a multi-level "Customer-Devices-Status" hierarchical structure. Figure 2b, above, illustrates various the data that may be available for each level of the structure. For example, the data that may be available at a Customer level may comprise industry, number of devices, device type distribution (such as wireless, router, switch, etc.), etc.; the data that may be available at a Device level may comprise device metadata, network topology, etc.; and the data that may be available at a Status level may comprise security issues, software versions, contract coverage, end of life issues, etc.

For a given customer, to find similar peers (each of which may have thousands of different devices) it may take three weeks on average for human beings to identify just a few similar peers, as people must manually compare and understand the complex relationship across devices and their status. At scale, where there may be thousands of customers, manually grouping the customers by considering all of the possible data elements that may make up a profile is practically impossible. Additionally, any comparisons that are made by hand will by definition be arbitrary and inconclusive (i.e., one is left not knowing what they do not know).

To group customers with large volumes of data and complicated information structures is not a trivial exercise. Basic and state of the art clustering algorithms (including, for example, k-means, hierarchical clustering, spectral clustering, Density-Based Spatial Clustering of Applications with Noise (DBSCAN), ordering points to identify the clustering structure (OPTICS), Balanced Iterative Reducing and Clustering using

Hierarchies (BIRCH), etc.) may be applied. However, the application any such algorithms will not provide an optimal clustering for a number of reasons. Such reasons include a complex hierarchy structure (e.g., the "Customer-Device-Status" structure that was described and illustrated above), thousands of relevant features with a correlation between same, thousands of potentially relevant and irrelevant features with or without correlations, the nonlinear nature of the related features, and the unpredictable nature of correlations and causality.

Accordingly, and as introduced previously, techniques are presented herein that support an intelligent system for grouping customers by leveraging exactly those things that make it impossible for a person to perform effective clustering. That is, the presented techniques support clustering based on a customer's vastly complicated network profiles. Aspects of the presented techniques encompass a smart customer grouping framework with highly accurate deep learning modeling, utilize domain-specific machine learning (ML) to unravel the nonlinear latent representations with a deep autoencoder, provide a flexible feature weighting capability to focus on certain features based on customer personas and business, and support a highly usable system in a production environment for sales and marketing professionals. Use of the presented techniques allows for the grouping of network customers, considering their complex hierarchical structure, using ML.

The next section of the instant narrative describes and illustrates a system architecture that supports aspects of the techniques presented herein and which may be referred to herein as a "Smart Grouping" system.

Figure 3, below, illustrates the main modules in a Smart Grouping system.

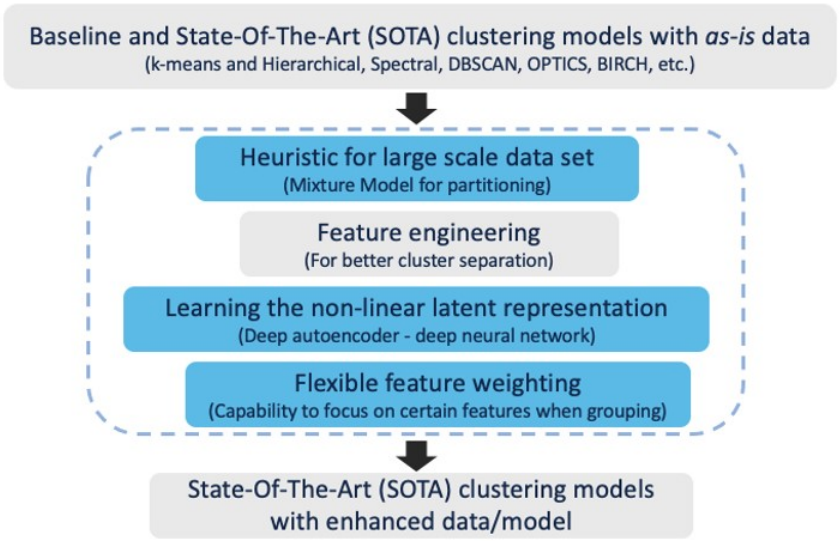

Figure 3: Exemplary Smart Grouping Modules

As depicted in Figure 3, above, a Smart Grouping system comprises several main steps (which are highlighted in blue in the figure) for yielding a scalable solution for large-scale network customers with complex hierarchical data. A first step encompasses a heuristic for large-scale data sets with mixture modeling for partitioning, a second step encompasses learning the nonlinear latent representation with a deep autoencoder, and a third step encompasses a flexible feature weighting capability to conditionally focus on certain features.

A first Smart Grouping module encompasses a heuristic for large-scale data sets using mixture modeling for partitioning.

Intuitively, the size of a customer's network is an important feature for grouping. For example, a large hospital with 10,000 devices and multiple campuses would not consider a small local hospital with 100 devices and a single campus to be their peer.

In statistics, a mixture model is a probabilistic model with a goal of representing the presence of subpopulations within an overall population. All of the data points are generated to form a mixture of a finite number of (unimodal) distributions. Expectation-maximization (EM), a maximum likelihood estimation, is seemingly the most popular technique used to determine the parameters of a mixture with an a priori given number of components and is very well suited for the instant problem. Aspects of the techniques presented herein leverage EM modeling to first separate the large scale customers by their

size to narrow the problem scope so that the model may focus on customers with a similar network size.

A second Smart Grouping module applies feature engineering for better cluster separation.

Feature engineering is one of the common practices in ML. Under aspects of the techniques presented herein, feature engineering aims to help the models focus more on the multimodal features by scaling and transforming the underlying feature space so that it can help the models discover the clustering structure. Aspects of the presented techniques rely on an automatic feature scaling through the variance of the feature itself, with min-max normalization employed in a working system that will be described and illustrated later in the instant narrative. Figure 4, below, depicts elements of the use of feature engineering to transform a data set.

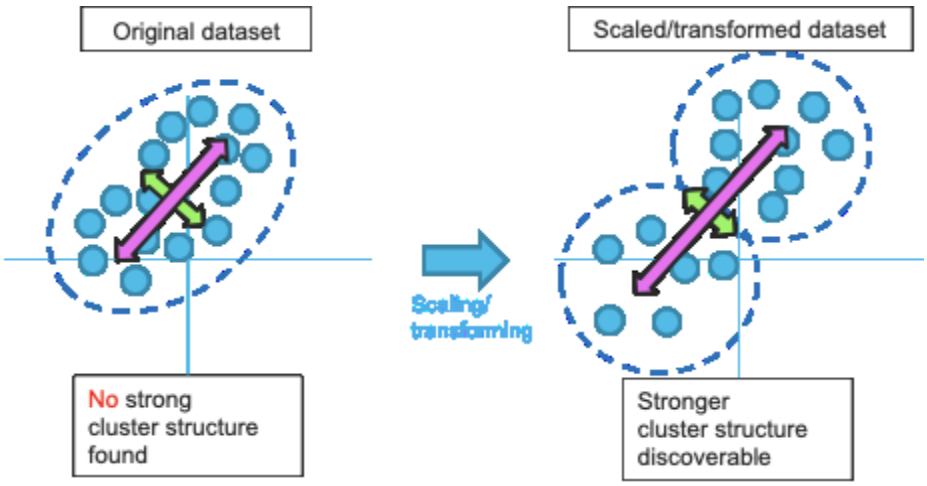

Figure 4. Exemplary Impact of Scaling/Transforming on Clustering

A third Smart Grouping module encompasses learning the nonlinear latent representation using a deep autoencoder.

An autoencoder is a type of neural network model which is trained to regenerate the input data with its output. An autoencoder has two parts – an encoder and a decoder. An encoder attempts to map the input data to a code layer and a decoder attempts to reconstruct the input data from the code layer. Additionally, the model is trained to reduce

the difference between the reconstructed output and the input. With a nonlinear encoder and decoder, autoencoders can learn the powerful nonlinear representation of the input.

Aspects of the techniques presented herein leverage an encoder that learns how to map the input to the learned feature (i.e., the code layer). The data that was described above, with a hierarchical structure, is very complex and requires a higher degree of effective nonlinear compression, which a deep autoencoder can produce.

Clustering methods for a flat data structure using deep neural networks have been widely studied. There are two approaches for combining deep neural networks with such clustering – sequentially applying clustering models and jointly optimizing both feature learning and clustering.

However, none of the existing deep learning clustering approaches can handle a hierarchical structure. As noted previously, the instant use case involves a hierarchical network structure encompassing a hierarchical information structure (e.g., the "Customer-Device-Status" structure that was described and illustrated above). Accordingly, aspects of the techniques presented herein support applying a deep autoencoder to hierarchically structured data.

A fourth Smart Grouping module encompasses personas-based flexible weighting.

Different personas (different people) may have diverse opinions regarding the importance of each level of information. Simply put, not all people will see the three levels of information (as described above) with the same importance. As a result, it is highly desirable to have a configurable mechanism. Figure 5, below, depicts elements of such a situation.

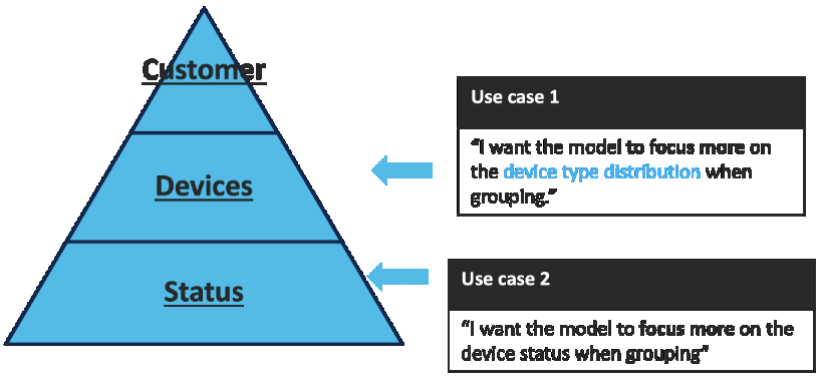

Figure 5: Flexible Feature Weighting Use Cases

As shown in Figure 5, above, one group of people may want a model to focus on device-level information while another group of people may want the model to focus on another level of information when the model performs a clustering.

Aspects of the techniques presented herein support a default (or suggested) group of settings which different personas can easily configure. By default, the model focuses more on the device-level information. This level contains information on the device type distribution within a network. For example, a customer whose network has a similar type of device distribution can see that they are creating and managing their network similarly when compared to dissimilar customers. Since the status of the devices may be similar, by chance, in the different networks, it makes sense to put more weight on the device type distribution.

Accordingly, aspects of the techniques presented herein support the flexibility or configurability to focus on a certain level in the hierarchical information structure (a capability that is new in the network domain for the peer comparison use case).

The next section of the instant narrative describes and illustrates various results that were obtained from a working system wherein the ML models, according to the techniques presented herein and as described above, were tested using over 3,000 instances of real customer data.

Within the working system, through heuristic partitioning the customers may first be divided into small, medium, and large network size groups based on the number of collected network devices, as shown in Figure 6, below.

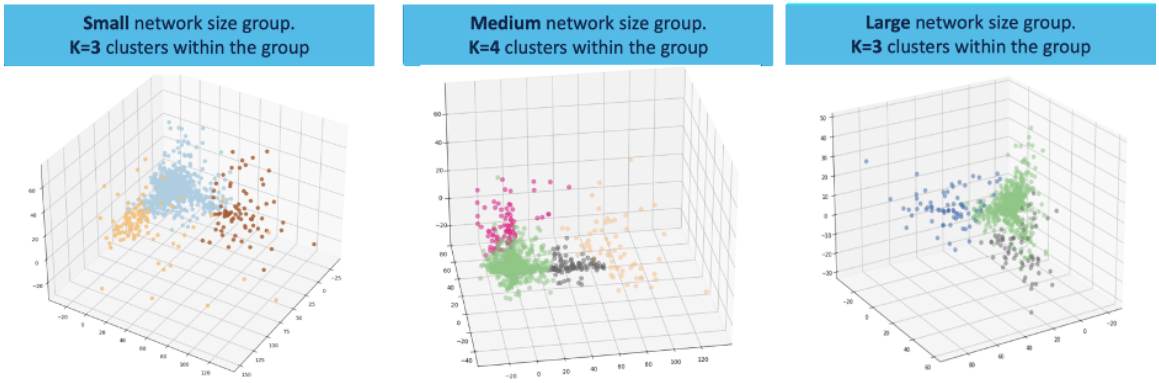

Figure 6: Heuristic Partitioning by Group Size

In each network size group, feature engineering, a deep autoencoder, and a default feature weight (with more focus on the device type distribution) were applied to group the data into N clusters.

The number of clusters, k is a hyperparameter which may be decided by performance evaluation metrics such as a silhouette score or coefficient. A silhouette value measures how similar a data point is to its assigned cluster (i.e., cohesion) compared to other clusters (i.e., separation). Such a value ranges from −1 to +1, where a high value indicates that the data point is well matched to its assigned cluster and poorly matched to neighboring clusters. The average over all of the data points summarizes how well the clustering was performed. In the working system the number of clusters for each network size group was selected where it maximized the average silhouette score or coefficient.

Figures 7a, 7b, and 7c, below, illustrate how a deep autoencoder can successfully reconstruct the input data using a code layer.

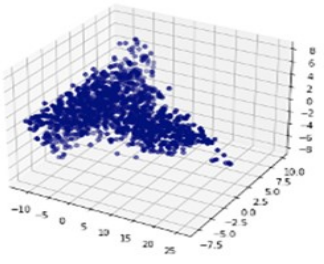

Figure 7a: Input Data

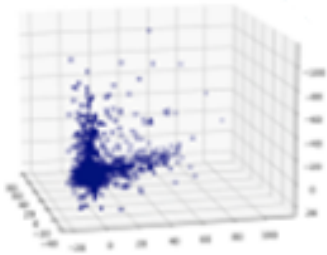

Figure 7b: Code Layer

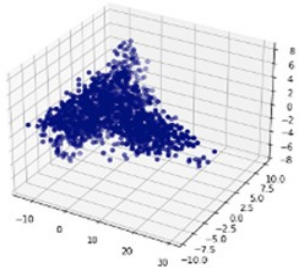

Figure 7c: Generated Data

Figure 7: Deep Autoencoder Visualization

Figure 7a, above, illustrates the input data (projected in three dimensions (3D)). Figure 7b, above, illustrates the encoded layer, depicting what the encoder learned from the input data. Figure 7c, above, illustrates the reconstruction of the input data using the code layer without any knowledge of the input data (i.e., the input data that is generated by the deep autoencoder). As depicted in the above figures, the reconstructed data and the input data demonstrate a high degree of similarity (i.e., the reconstructed input data that is generated by the deep autoencoder shows a high similarity with the input data).

Additionally, the code layer is in a much smaller dimension so the deep autoencoder was forced to learn how to compress complex data.

Among other things, the working system was validated at a group level. To verify the model results, features may be compared in groups, where customers in the same cluster should show similar features while customers in different clusters should show different features. Figures 8a and 8b, below, illustrate various results for three groups in a "large" size family.

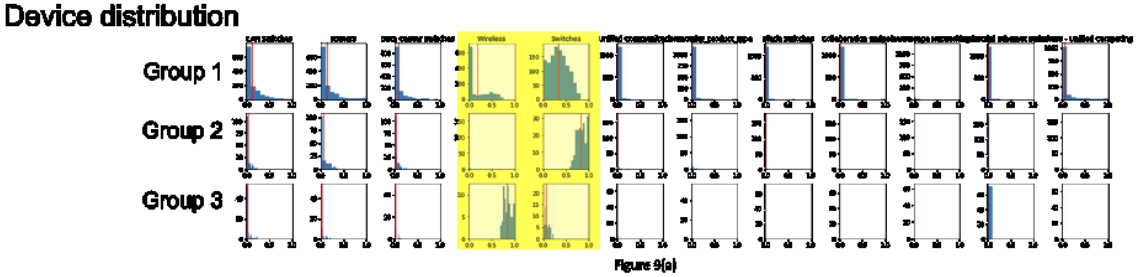

Figure 8a: Model Validation at Group Level – Device Type

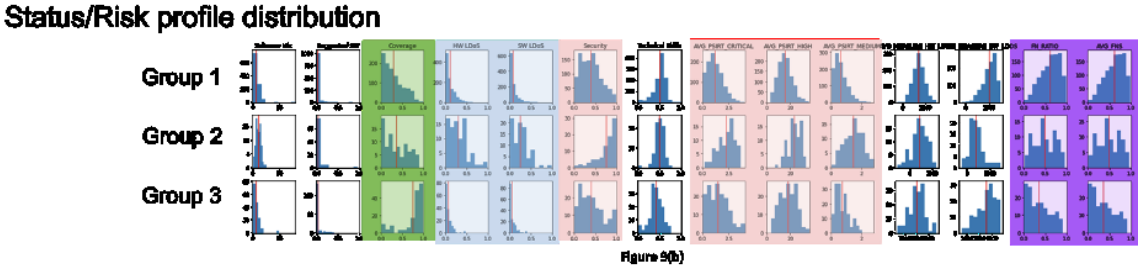

Figure 8b: Model Validation at Group Level – Status/Risk

As depicted in the above figures, the model can successfully cluster similar customers, in terms of device type distribution, into the same group. For example, in Figure 8a, above, all of the customers in Group 2 show a heavy “Switches”-based network (e.g., a large percentages of their devices are of device type switch) while showing almost zero, to just a few, wireless devices. On the other hand, all of the customers in Group 3 exhibit a heavy “Wireless”-based network while showing very small percentages in the “Switches” category.

Similarly, Figure 8b, above, displays the histogram of the status (risk profile) feature. Depicted are three groups and their risk profiles with one status or risk component in each column. Highlighted are the many risk components which show very different distribution shapes.

In summary, the model groups similar customers in terms of the device type distribution and their device status or risk profile into the same cluster. While doing this, the characteristics of each group may be discovered. For example, Group 2 is characterized, as mentioned above, with a "Switches"-heavy network with very few "Wireless" devices and with high risk status/profiles.

Additionally, the working system was validated at a customer level. For each group, a customer may be randomly selected and queried. Then, several customers may be selected from the same group and several customers may be selected from a different group. The goal is to see if the customers from the same group exhibit features that are similar to customers in the same group and are different from customers in the other groups.

The Figure 9, below, displays the device type distribution feature for a query customer (left panel), customers from the same cluster (center panel), and customers from another cluster (right panel).

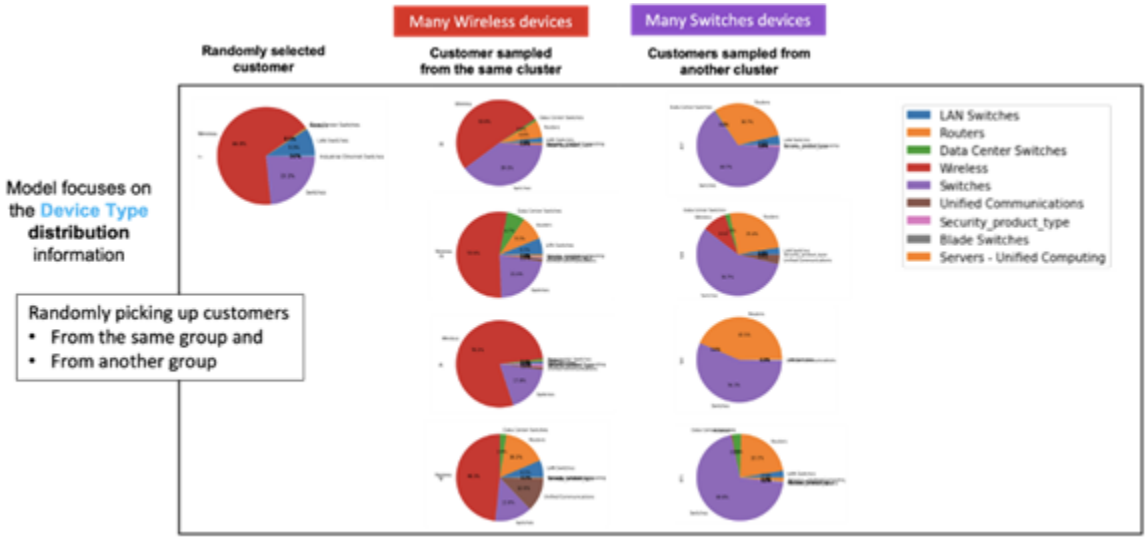

Figure 9: Model Validation at Customer Level – Device Type

As depicted in Figure 9, above, the query customer has many "Wireless" device types (i.e., they are a heavy "Wireless"-based network). Customers from the same group are also heavy "Wireless"-based networks whereas customers from another group are heavy "Switches"-based networks.

Similarly, Figure 10, below, displays the status (risk profile) feature for a query customer (shown in the left panel), customers from the same cluster (shown in the center panel), and customers from another cluster (shown in the right panel).

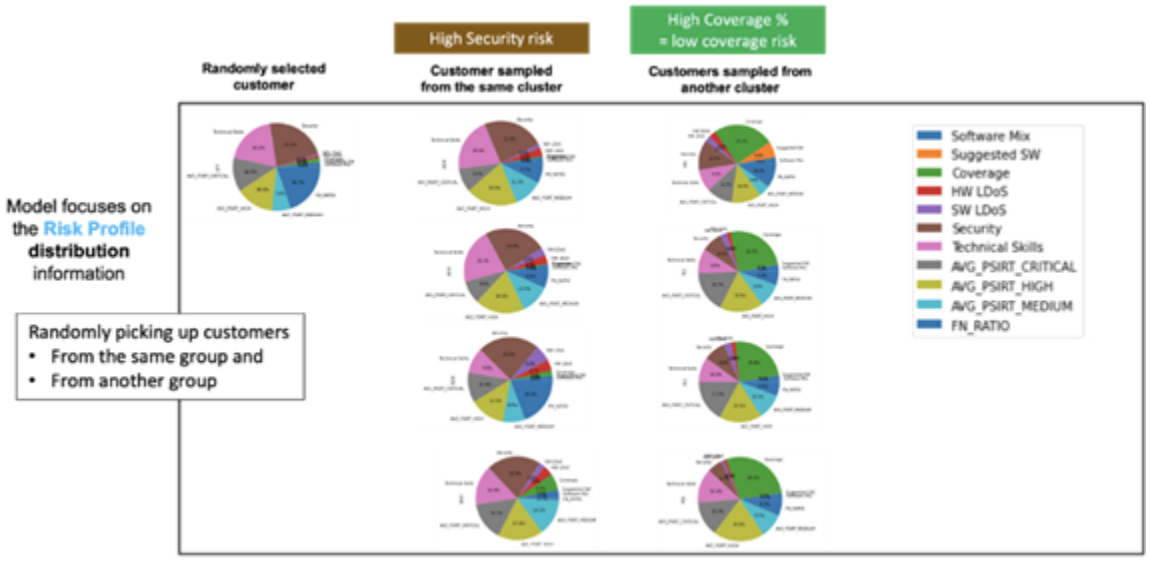

Figure 10: Model Validation at Customer Level – Status/Risk

As depicted in Figure 10, above, the query customer has high "Security" related risks. Customers from the same group also show high "Security" risks whereas customers from another group show low security risks and low coverage risks.

Finally, the working system was validated using performance metrics (e.g., a silhouette score or coefficient). As shown in Figures 11a and 11b, below, the average silhouette score shows a significant improvement over the native clustering with the as-is data.

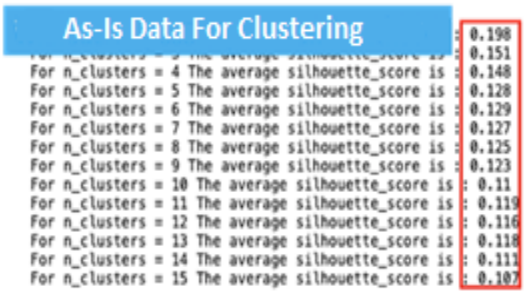

Figure 11a: Before (As-Is Data)

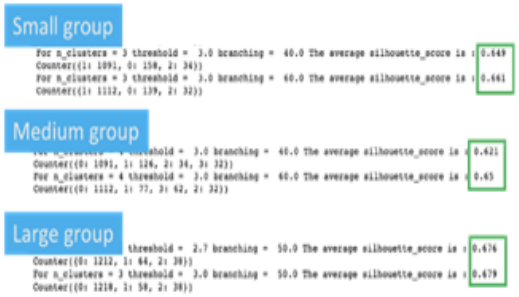

Figure 11b: After (Model Improvement)

Figure 11a, above, illustrates the baseline of applying a clustering model (employing k-means) to the original feature data while Figure 11b, above, depicts the results from applying the same models to the improved data set.

Among other things, the working system validates that the application of aspects of the techniques presented herein can significantly improve the model from 0.2 (in the baseline) to 0.6. Such an improvement stems from the use, according to the presented techniques, of heuristic portioning, feature engineering, and a deep autoencoder.

In summary, a Smart Grouping system according to aspects of the techniques presented herein can successfully partition a complex network hierarchy structure using deep learning techniques.

In addition to supporting the ML model testing that was described above, the working system that was noted previously also incorporated a user interface. Figure 12, below, illustrates one particular display from that system.

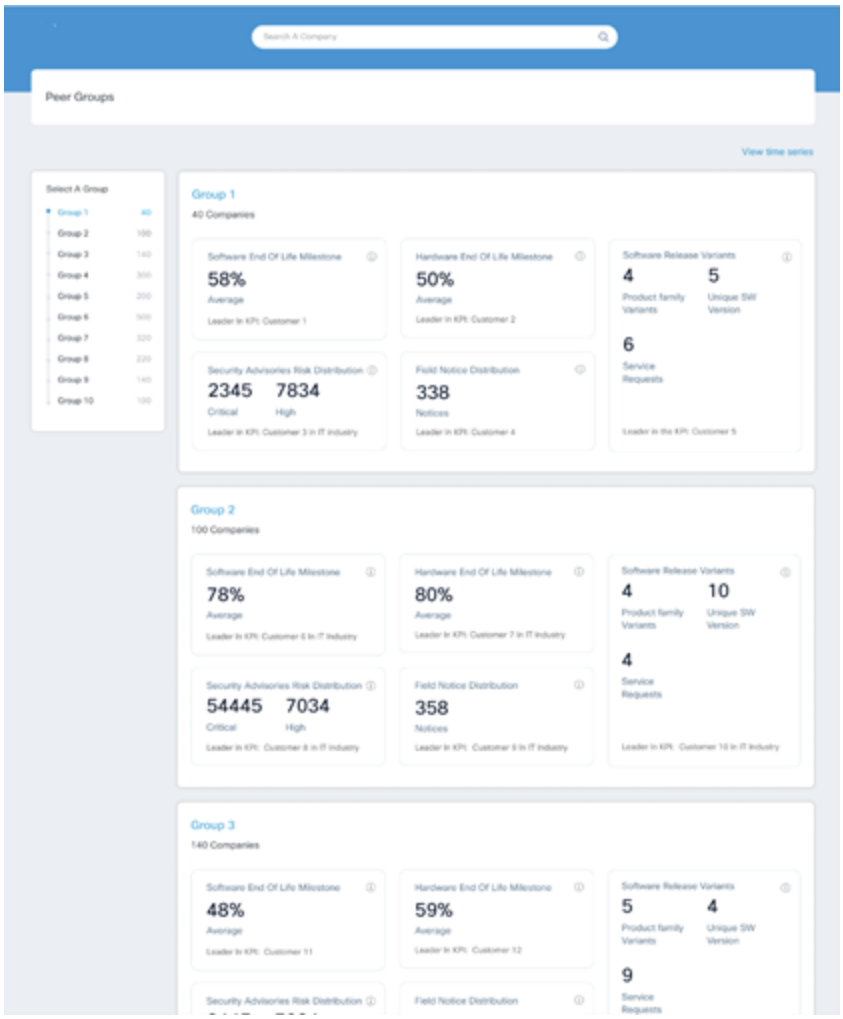

Figure 12: Group Customers Based on Hierarchical Complex Data

As depicted in Figure 12, above, the system provides an overview of customer grouping based on the Smart Group models that were described above. If a user is interested in any specific group, they may select that group and obtain more information, as shown in Figure 13, below.

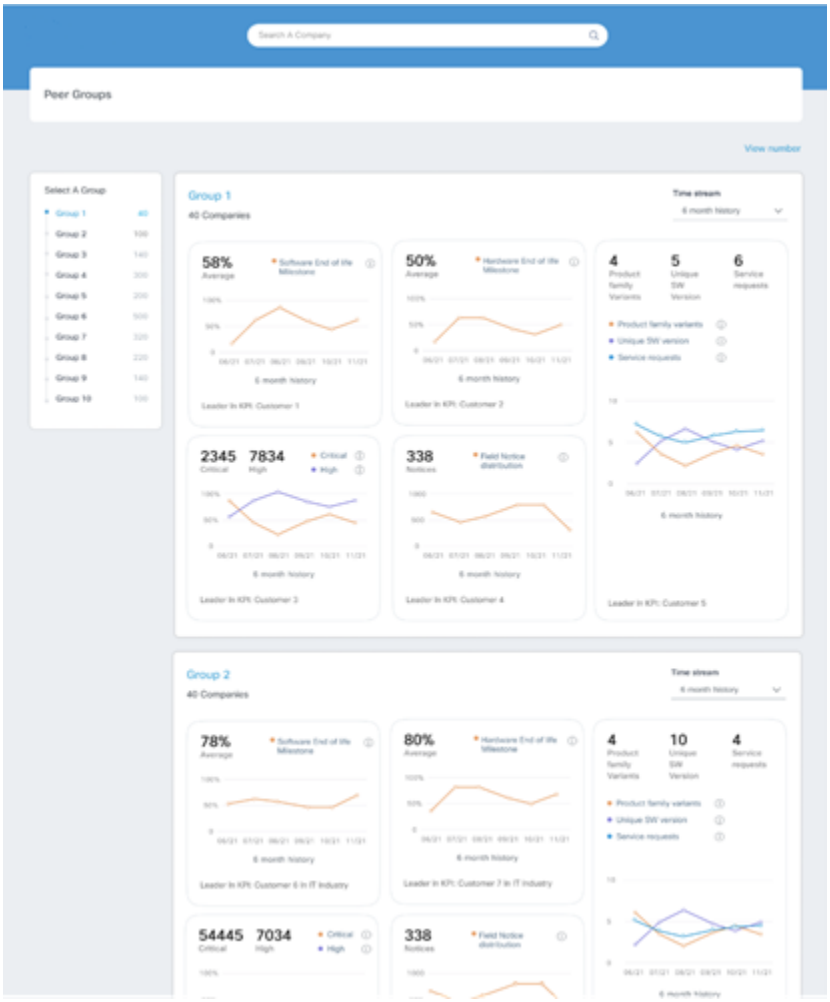

Figure 13: Exemplary KPIs and Benchmarking Values

Figure 13, above, illustrates the display of various key performance indicators (KPIs) and benchmarking values, including top performances for each KPI (e.g., security, aging, defects, etc.).

By employing the displays that were described and illustrated above, sales and marketing professionals can easily understand a customer's positions compared to peers in the same group, as shown in Figure 14, below.

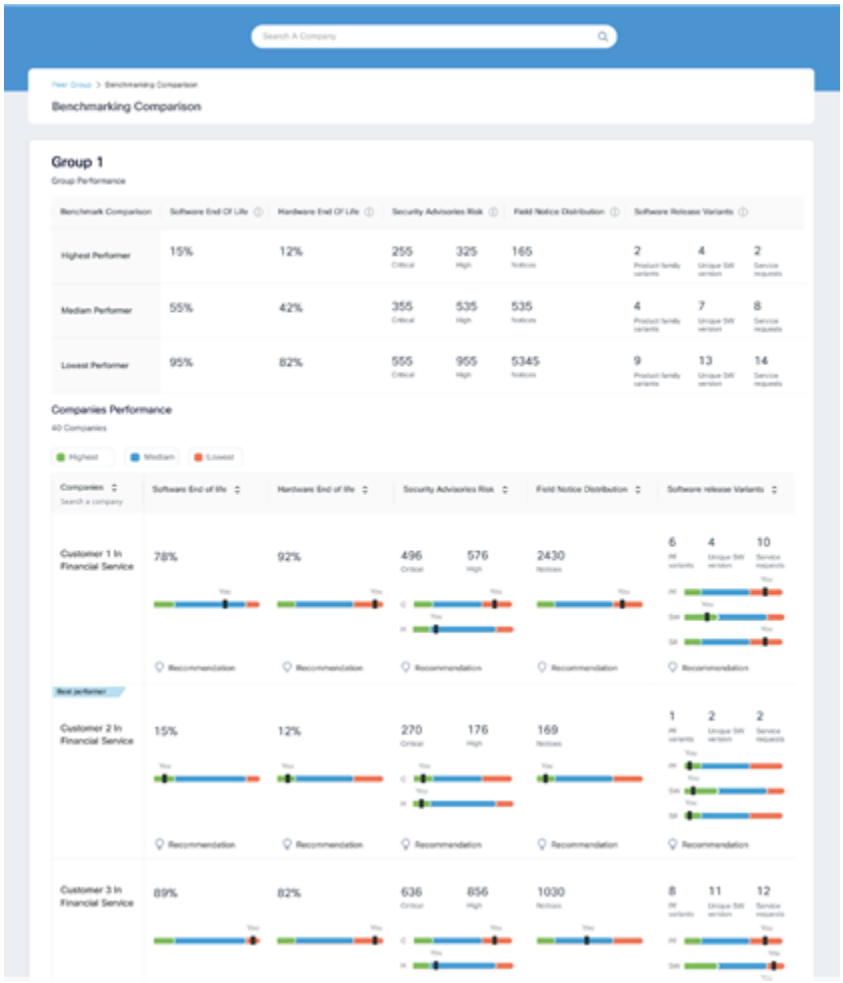

Figure 14: Exemplary KPIs and Benchmarking Values

Additionally, as depicted in Figure 15, below, the system provides actionable insights that may be proposed to each customer to improve their performance.

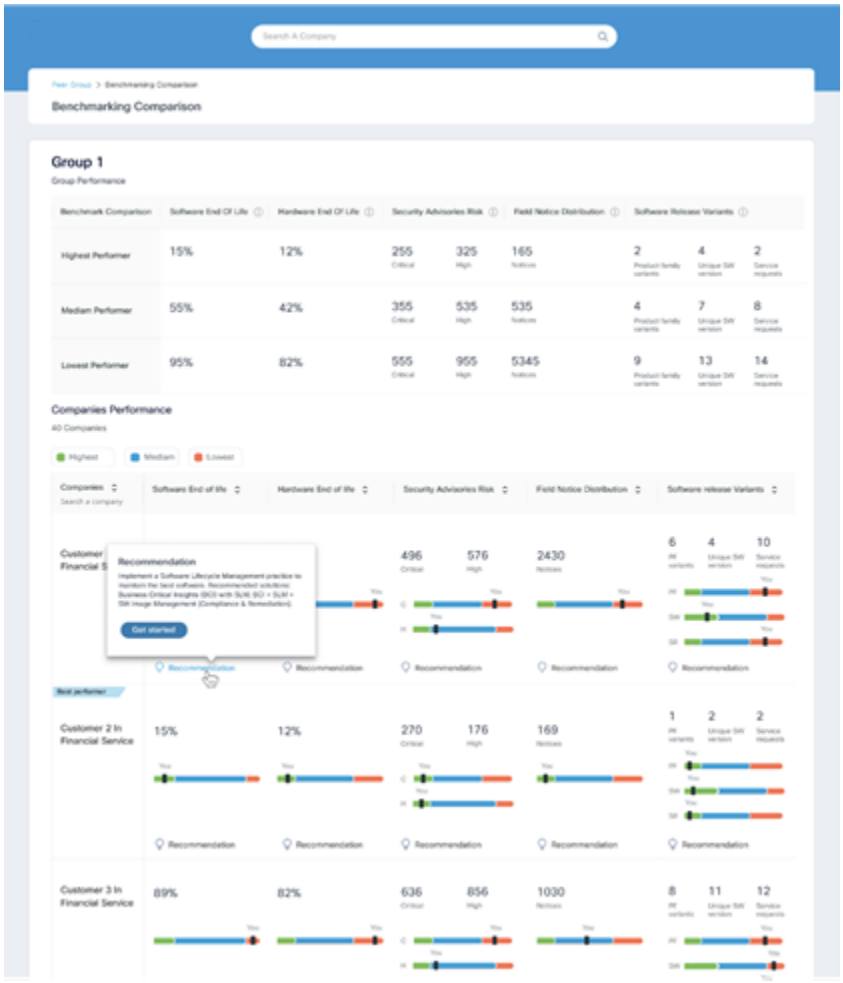

Figure 15: Actionable Insights and Recommendations

The information that was described and illustrated above enables a professional user to perform analyses that they never could before, allowing them to finally know the information that they did not previously know. Or in the words of Confucius, they may achieve 'true knowledge.'

In summary, techniques have been presented that support an intelligent system for grouping customers by leveraging exactly those things that make it impossible for a person to perform effective clustering. That is, the presented techniques support clustering based on a customer's vastly complicated network profiles. Aspects of the presented techniques encompass a smart customer grouping framework with highly accurate deep learning modeling, utilize domain-specific ML to unravel the nonlinear latent representations with a deep autoencoder, provide a flexible feature weighting capability to focus on certain

features based on customer personas and business, and support a highly usable working system in a production environment for sales and marketing professionals. Use of the presented techniques allows for the grouping of network customers, considering their complex hierarchical structure, using ML.