Optimizing Similar Audience Search in Targeted Advertising: Effectiveness of Siamese Networks for Autoencoder-based User Embeddings

Il'murat Tokhtakhunov
International Information Technology University, Manasa, Almaty, Kazakhstan
ilmurat.tokhtakhunov@gmail.com

Aizhan Altaibek
International Information Technology University, Manasa, Almaty, Kazakhstan | Institute of Ionosphere, Gardening Community IONOSPHERE 117, Almaty, Kazakhstan
a.altaibek@iitu.edu.kz

Marat Nurtas
International Information Technology University, Manasa, Almaty, Kazakhstan | Institute of Ionosphere, Gardening Community IONOSPHERE 117, Almaty, Kazakhstan
m.nurtas@iitu.edu.kz (corresponding author)

Received: 10 February 2025 | Revised: 12 March 2025 and 26 March 2025 | Accepted: 29 March 2025

 DOI: https://doi.org/10.48084/etasr.10527

ABSTRACT

This study investigates the effectiveness of using Siamese networks for comparing embedding vectors that describe user profiles. A model was developed to identify similar audiences in the context of targeted advertising. The analysis of the requirements for such a model revealed that traditional approaches to tabular data processing often struggle to address the unique challenges posed by this task, particularly in terms of scalability and adaptability. The proposed approach allows for the effective identification of lookalike users without relying on explicit feature engineering. This method was evaluated using an anonymized proprietary dataset provided by a telecommunications operator, which included sociodemographic descriptions of subscribers, their tariff plans, and mobile devices. Experimental results showed that the model achieved an F1 score of 0.75, a ROC-AUC of 0.79, and a lift score in the top 1 of 12.9, outperforming baseline methods in targeted user identification by 41.61% on average. The results highlight the ability of the proposed method to meet the key requirements for this task, showcasing its effectiveness and scalability. This study highlights the versatility of the proposed approach, emphasizing its applicability across various domains for tabular data classification tasks. Future research will focus on developing multiple autoencoders tailored to different domains and integrating them to solve specific tasks.

Keywords-user profiling; siamese network; embeddings; audience selection; autoencoder; targeted advertising; cosine similarity distance

I. INTRODUCTION

In recent years, the rapid development and adoption of machine learning methods has reached significant scales. Companies across various industries actively invest in these technologies, recognizing their potential to increase revenues and optimize costs. Alongside this trend, advances in big data infrastructure reflect the growing emphasis on effective data management [1]. Organizations retain the data they generate, leading to its continuous growth and an increasing number of features for machine learning analysis. However, an increase in the number of features introduces challenges associated with high-dimensional models, commonly referred to as the curse of dimensionality. This issue, characterized by data sparsity in high-dimensional spaces, often reduces a model's ability to effectively identify meaningful patterns [2]. Data sparsity can lead to decreased model performance and a high risk of overfitting. This phenomenon complicates the identification of similar users, as traditional methods struggle to distinguish meaningful relationships in high-dimensional representations.

Traditional feature engineering requires significant expertise and effort, which becomes impractical with high-dimensional data. As data dimensionality increases, manual feature engineering becomes challenging [3]. Traditional similarity detection methods that rely on distance calculations in multidimensional spaces perform well for images and text, due to their inherent suitability for vectorized representations. Autoencoders, a type of neural network capable of generating compressed data representations, are employed to address the challenge of high dimensionality, create embedding vector representations of tabular data, and improve model performance. Autoencoders facilitate unsupervised learning by encoding data into lower-dimensional spaces, enabling better identification of complex patterns and nonlinear relationships. These representations are highly informative and help reduce data sparsity [4]. Moreover, these representations have been proven to be more effective compared to the initial tabular format when used with algorithms to compare and search for similar embedding representations. In this domain, Siamese networks have gained significant popularity, primarily in tasks that involve the search for similar images and texts. Siamese networks are a type of neural network designed to determine the similarity between two inputs by comparing their feature representations. These networks consist of two or more identical subnetworks that share the same weights and structure, ensuring consistent feature extraction from inputs. Typically, the outputs of these subnetworks are combined using a similarity function, such as the Euclidean distance or cosine similarity, to quantify the relationship between the inputs. Siamese networks are widely used in applications such as image recognition, text similarity, and one-shot learning, due to their ability to effectively measure similarity in high-dimensional spaces.

This study identified a practical task: the development of a universal look-alike model for automated identification of similar users within the context of targeted advertising. The primary goal was to create a high-performance and versatile approach to minimize the analysts' workload. The model was designed to accept a reference sample of users and identify similar users within a larger database. The key criteria for the model were high speed, accuracy, adaptability, and scalability. The architecture of the applied auto-encoders was tailored to specific data features from the telecommunications industry. The autoencoders integrate feature vectors from various entities (e.g., subscribers, devices, base stations), enhancing the model's ability to account for complex interdomain relationships [5].

The model was evaluated using standard binary classification metrics (lift-top1, precision, recall, F1-score, ROC AUC) as well as business metrics, such as advertising campaign conversion rates. Validation was carried out on data from 17 different advertising campaigns. For each campaign, 90,000 subscribers with the highest similarity scores were selected. All features were calculated during the monthly subscriber base feature collection process. These subscribers received advertising SMS messages containing a link to the promoted product of the B2B customer. These pilot campaigns ensured an objective evaluation of the proposed approach.

The shift from the traditional approach of training a separate binary classifier for each advertising campaign was driven by the need to optimize analysts' time and resources [6]. The objective was to develop a universal look-alike search model that could function as an automated service. This service would take a reference sample of users as input and efficiently identify others with similar characteristics within a shared database, streamlining the process and improving scalability. In real-world applications, this service would enable B2B clients to access an online platform, upload their current target audience as a positive seed for the look-alike model, specify the desired size of the subscriber sample for advertising campaigns, complete the payment for the targeted audience selection, and, following the look-alike scoring process, download the generated subscriber sample containing the identified target audience. All of these processes will be fully automated, eliminating the need for the assistance of a business development manager or a data analyst.

The effectiveness of such a model depends on the availability of a comprehensive and diverse set of features for evaluation [7]. A truly flexible look-alike search model must be capable of seamlessly adapting to any dataset or task without relying on predefined task-specific features. Furthermore, the envisioned service was required to simultaneously accommodate a large volume of B2B clients, ensuring rapid and efficient request processing while minimizing task backlogs. Consequently, achieving high-speed performance emerged as a critical objective. Unfortunately, conventional approaches were unable to provide a solution that successfully balanced the competing demands of speed, accuracy, and feature diversity.

II. AUTOENCODER, DATA PREPARATION, AND TRAINING

The proprietary training dataset was obtained through a collaboration agreement with a telecommunications operator. Since this task originates from a real-world business case, one of the authors, working as a data scientist in the big data department, received a request from the business team to develop a flexible look-alike model. The dataset was compiled from existing datamarts within the company that generate user profiles, incorporating various available data sources to create a more comprehensive and multidimensional dataset. The data was fully anonymized in strict compliance with the company's fairness policy and relevant legislative regulations to ensure privacy protection. Due to confidentiality restrictions, the dataset is not publicly available. However, access was granted under a research partnership, ensuring that all data processing adhered to strict privacy protection protocols.

The dataset comprises detailed information about 900,000 subscribers organized into 948 features grouped into domains referred to as entities. The User Entity offers an in-depth profile of each subscriber, including aggregated metrics on activity, traffic usage, interests, and sociodemographic characteristics. The Web Entity captures subscribers' online behavior and points of interest, derived exclusively from anonymized Internet sessions collected with explicit user consent. The Finance Entity integrates banking transaction data to create personalized shopping baskets and customer profiles

based on users' payment histories and purchasing behavior. The Device Entity provides descriptions of subscribers' mobile devices, including technical specifications, price segmentation, and prevalence across various demographic groups. The Cell Base Station Entity details the primary base stations used by subscribers, consolidating information on service quality, network stability, internet speed, and user geoactivity. Finally, the Tariff Plan Entity delivers a comprehensive analysis of each subscriber's tariff plan, covering its features, pricing, and overall popularity among the user base. The dataset contained both quantitative and qualitative data. Since subsequent methods such as autoencoders and Principal Component Analysis (PCA) require numerical input, all qualitative data were encoded using the one-hot encoding technique. The preprocessing steps, including data normalization and addressing missing values, were critical to preparing the dataset for effective autoencoder training. This preprocessing involved addressing multicollinearity, imputing missing values, scaling, and performing other necessary transformations. The pairwise correlation coefficient method was employed to mitigate the issue of linearly dependent variables. This technique evaluates the degree of linear correlation between two features. When the correlation coefficient between two features exceeded a predefined threshold, the less informative variable was removed. The threshold value was determined through iterative experimentation, exploring a range of values between 0.5 and 1. Various models were built, and their performance was evaluated. Through iterative testing, a threshold of 0.77 was identified as optimal, offering a balanced trade-off between precision, recall, and model stability for this dataset. Consequently, this value was selected as the final correlation threshold. By removing variables that exceeded this threshold, redundancy was minimized and the effects of linear dependence were mitigated, contributing to the overall efficiency of the model [8].

The handling of missing values posed a significant challenge in several fields within the dataset. A range of imputation techniques was employed to address it. Although more advanced methods such as Multiple Imputation by Chained Equations (MICE) were considered, they did not yield significant performance improvements compared to simpler approaches. Furthermore, given the large size of the dataset, these complex methods were computationally intensive and time-consuming, making them impractical for production environments where efficiency is crucial [9]. Instead, simpler statistical methods, such as mean, median, and mode imputation, were used alongside random value generation to enhance robustness. Each method was carefully applied based on the characteristics of the missing data, ensuring a thorough and efficient imputation process while maintaining the dataset's integrity for subsequent modeling [10].

In addition to addressing multicollinearity and imputing missing values, the data set was further refined using isolation forests, an anomaly detection technique derived from Random Forests (RF). Isolation forests operate by constructing decision trees to isolate data points that deviate significantly from the majority. Anomalies are identified by their shorter path lengths in the trees, reflecting their relative isolation. This method proved highly effective in identifying and removing unusual patterns within the training dataset, ensuring that the model performance was not compromised by atypical data points. By filtering out outliers, isolation forests contributed significantly to enhancing the model's robustness and reliability.

Another critical preprocessing step was data scaling, which ensured uniformity across features and mitigated potential biases during the model training process. Scaling is particularly vital for training autoencoders, which are sensitive to the magnitude of input features [11]. Without proper scaling, features with higher numerical values could dominate the learning process, diminishing the contribution of features with lower values. Given the heterogeneity of the input features, min-max normalization was applied to rescale all numerical features to the [0,1] range. Normalization was applied to ensure that all features contribute equally to the model, preventing the dominance of variables with larger magnitudes. Categorical features were transformed using one-hot encoding, allowing the model to leverage categorical relationships effectively. The entire preprocessing pipeline was applied consistently to both training and validation datasets to maintain data integrity and avoid distribution shifts. By normalizing all features to a consistent scale, the autoencoder was able to focus equally on all input dimensions, effectively reconstructing the data and preserving its inherent structure.

After completing all preprocessing steps, including outlier detection, imputation, and scaling, the dataset was thoroughly refined, resulting in a cleaner, more balanced input ready for training the autoencoder. These steps ensured that the data were optimized for the subsequent modeling process, laying the foundation for accurate and efficient learning [12]. An autoencoder consists of two fundamental components: the encoder and the decoder. The encoder processes the input feature vector, transforming it into a hidden lower-dimensional embedding. This component is responsible for efficiently extracting the most significant features from the input data, reducing its dimensionality while retaining the essential structure and characteristics of the original features [13]. The decoder, in contrast, reconstructs the original feature vector from the hidden embedding produced by the encoder. Its primary function is to recreate the input data from the compressed representation while preserving meaningful patterns and features. During the training process, the autoencoder minimizes the reconstruction error, enabling the decoder to accurately capture the critical characteristics of the original data [14].

In the context of this task, transforming the input sparse representation into a compressed dense embedding representation was critical due to several valuable properties:

- Compactness: The compressed representation vector typically has much lower dimensionality than the original data, significantly reducing the number of parameters and accelerating the model training process.
- Semantic similarity: The vector representation obtained through embedding accounts for the semantic relationships between objects, allowing for the identification of similar objects within the compressed representation space [15].

- Generalization capability: the compressed vector can encapsulate generalized features of objects, making it suitable for various tasks such as classification, regression, and clustering.

To ensure optimal performance, a stacked autoencoder was designed with multiple layers in both the encoder and the decoder. The encoder consists of two fully connected layers, each followed by batch normalization and LeakyReLU activations. The decoder mirrors this structure with an additional output layer that reconstructs the original input features. The latent space dimension was set at 288, ensuring a balance between compression and representational capacity. The LeakyReLU activation function was chosen instead of standard ReLU to mitigate the problem of dying neurons, ensuring that even small gradient values propagate during training. This activation function helps maintain stable weight updates, improving the model's ability to learn meaningful embeddings.

The model was trained using the Adam optimizer due to its adaptive learning rate mechanism and its ability to handle sparse gradients effectively. Compared to Stochastic Gradient Descent (SGD), Adam provides faster convergence while maintaining robust generalization properties. The learning rate was initially set at 0.001, with a decay schedule applied to gradually reduce it during training, preventing divergence and improving the stability of the final model. Hyperparameter tuning was performed using grid search, optimizing key parameters such as learning rate (set at 0.001), batch size (set at 512), and L2 regularization strength (set at 0.0001). A batch size of 512 was chosen to balance computational efficiency and convergence stability. L2 regularization (weight decay of 0.0001) was applied to improve generalization and prevent overfitting in the high-dimensional feature space.

The Mean Squared Error (MSE) was used as the loss function for optimization, as it is well-suited for reconstruction tasks where the goal is to minimize the difference between input and output feature vectors. The model was trained for 400 epochs, with training and validation losses monitored at each step. Although early stopping was considered to prevent overfitting, the final model selection was primarily based on its performance on unseen advertising campaign data, rather than solely on validation loss stability. This approach ensured that the model generalized well to real-world applications while maintaining high reconstruction fidelity. This architecture and training strategy allowed the autoencoder to effectively extract compact and meaningful representations while maintaining high reconstruction accuracy and computational efficiency.

After numerous iterations of testing and analysis, the final encoder model architecture was chosen. Figure 1 presents the detailed architecture of the final Stacked Auto-Encoder (SAE) model used in this study. The model consists of an encoder, a decoder, and a latent representation layer. The encoder includes two hidden layers of 1000 neurons each, using LeakyReLU activations and batch normalization to enhance training stability and convergence. The decoder mirrors this structure in reverse, reconstructing the original 948-dimensional input from the compressed 288-dimensional latent space. The multi-layered structure allows for deeper feature extraction, allowing the model to learn complex patterns in the data. By incorporating batch normalization into each layer, training stability and convergence speed were improved, reducing the risk of vanishing or exploding gradients. The LeakyReLU activation function was chosen to prevent neuron deactivation, ensuring that all layers contribute to the representation learning process. The latent representation obtained from the encoder serves as the basis for the similarity estimation process in the Siamese network.

```
StackedAE(
  (encoder): Encoder(
    (net): Sequential(
      (0): Linear(in_features=948,
                  out_features=1000,
                  bias=True)
      (1): BatchNorm1d(1000,
                       eps=1e-05,
                       momentum=0.1,
                       affine=True,
                       track_running_stats=True)
      (2): LeakyReLU(negative_slope=0.2, inplace=True)
      (3): Linear(in_features=1000,
                  out_features=1000,
                  bias=True)
      (4): BatchNorm1d(1000,
                       eps=1e-05,
                       momentum=0.1,
                       affine=True,
                       track_running_stats=True)
      (5): LeakyReLU(negative_slope=0.2, inplace=True)
    )
  )
  (decoder): Decoder(
    (net): Sequential(
      (0): Linear(in_features=288,
                  out_features=1000,
                  bias=True)
      (1): BatchNorm1d(1000,
                       eps=1e-05,
                       momentum=0.1,
                       affine=True,
                       track_running_stats=True)
      (2): LeakyReLU(negative_slope=0.2, inplace=True)
      (3): Linear(in_features=1000,
                  out_features=1000,
                  bias=True)
      (4): BatchNorm1d(1000,
                       eps=1e-05,
                       momentum=0.1,
                       affine=True,
                       track_running_stats=True)
      (5): LeakyReLU(negative_slope=0.2, inplace=True)
      (6): Linear(in_features=1000,
                  out_features=948,
                  bias=True)
    )
  )
  (latent_layer): Linear(in_features=1000,
                         out_features=288,
                         bias=True)
)
```

Fig. 1. The final architecture of the autoencoder.

Additionally, the latent space dimension was carefully selected to balance information retention and computational efficiency. A larger latent space allows for richer feature representation but increases the risk of overfitting, whereas a

smaller latent space enhances generalization but may lead to information loss. The chosen latent space of 288 dimensions provided an optimal trade-off, as confirmed through experimental validation. Overall, this architecture enables the model to generate robust user embeddings, improving similarity search accuracy while maintaining computational efficiency.

III. APPLICATION OF SIAMESE NETWORKS

Siamese networks are a specialized type of neural network architecture designed for similarity measurement and comparison tasks. Unlike conventional neural networks, which are trained to classify inputs, Siamese networks focus on learning a similarity function between input pairs. This unique design makes them particularly effective for applications where identifying relationships between objects, rather than their classification, is the primary goal. The core architecture of a Siamese network consists of two or more identical subnetworks that share weights and parameters. Each subnetwork processes one of the inputs in a pair and extracts a feature representation in the form of an embedding vector [16]. These embedding vectors are then compared using a similarity function to determine the degree of resemblance between the inputs. Siamese networks map high-dimensional inputs to a shared feature space, enabling the identification of relationships with greater precision [17].

This study used Siamese networks to identify and compare compressed embedding representations derived from tabular data. Although Siamese networks have traditionally been applied to tasks such as image recognition, face verification, and natural language processing, their integration with autoencoders allows for novel applications in domains such as telecommunications [18]. The use of Siamese networks enhances the ability to compare users by evaluating the similarity of their dense representations in a high-dimensional feature space. This approach offers several advantages. First, it provides a robust mechanism to capture semantic similarities within complex datasets, enabling the accurate identification of related objects or users. Second, the shared architecture of Siamese networks ensures consistency and efficiency in processing data pairs, even in large-scale datasets. Finally, the combination of Siamese networks with dense embedding representations generated by autoencoders mitigates the challenges associated with sparsity and dimensionality in traditional tabular data analysis [19]. By integrating Siamese networks into the modeling pipeline, this study demonstrates their potential for advanced similarity measurement, opening new opportunities for their application in various industries and research domains [20].

Training Siamese networks required the pairing of similar and dissimilar users, which was achieved through a Cartesian product for positive pairs and random sampling for negative pairs [21]. This pairing was performed according to the validation task. For positive pairs, all users included in the positive seed, which represents the reference audience for which similar users were to be identified, were used to form pairs of similar users through a Cartesian product [22]. This approach ensured that all relevant similarities available were fully utilized. For negative pairs, a random negative sampling strategy was applied, where each user in the positive seed was randomly paired with a user from the remaining subscriber base, ensuring that they belonged to different target groups. This method was chosen because random negative sampling introduces a diverse set of dissimilar pairs, allowing the model to learn a wider range of variations in user profiles and improving its ability to generalize. Alternative approaches such as hard negative mining were considered but were found to be less effective in the early training stages, as they could lead to unstable decision boundaries before the model had developed a stable feature space. The training process followed the commonly accepted class distribution for contrastive learning tasks, with 10% of the dataset consisting of positive pairs and 90% consisting of negative pairs. This balance ensures that the model does not overfit positive examples while maintaining an effective decision function for similarity estimation.

The computational complexity of training a Siamese network is significantly higher compared to traditional models such as Support Vector Machines (SVM) and RF. Unlike these classical methods, which operate on individual samples, Siamese networks require processing paired inputs, effectively increasing the computational cost by a factor of the number of training pairs. The training complexity of SVM and RF largely depends on the number of samples and feature dimensionality. In contrast, the complexity of a Siamese network grows with the number of pairs and the depth of the neural network, making it more computationally demanding. Neural networks require GPU acceleration for efficient training, whereas classical methods can be trained on standard CPUs with lower memory requirements. Despite the increased training cost, the inference phase of the Siamese network remains efficient, as once the model is trained, user embeddings are precomputed and stored, allowing for fast similarity calculations during real-time look-alike audience searches. This trade-off makes the approach computationally viable for large-scale applications where high accuracy is a priority.

After a series of extensive experiments with various Siamese network architectures, their performance was examined using multiple evaluation metrics, including precision, recall, F1-score, and ROC AUC. The optimal neural network structure was identified through empirical testing, demonstrating superior effectiveness in capturing meaningful similarities between users while maintaining computational efficiency. Figure 2 illustrates the finalized architecture of the Siamese neural network used to evaluate user similarity. Each twin network receives a 288-dimensional input vector - a user embedding obtained from the autoencoder - and processes it through five fully connected layers with batch normalization and ReLU activation. Intermediate layers consist of 256, 512, 224, 160, and 128 units, with a dropout layer added before the final dense layer to prevent overfitting. The two parallel branches share weights and output fixed-length embeddings, which are then compared using a similarity distance function (e.g., Euclidean or cosine) [23]. The result of this comparison is used in the loss function to guide the learning process, optimizing the network to correctly distinguish between similar and dissimilar user pairs. This architecture plays a central role in the method, allowing the model to learn high-level semantic similarities between users without explicit feature engineering.

Figure 2 shows how the embedding space is aligned and optimized through training, directly supporting the core objective of improving look-alike audience identification in targeted advertising.

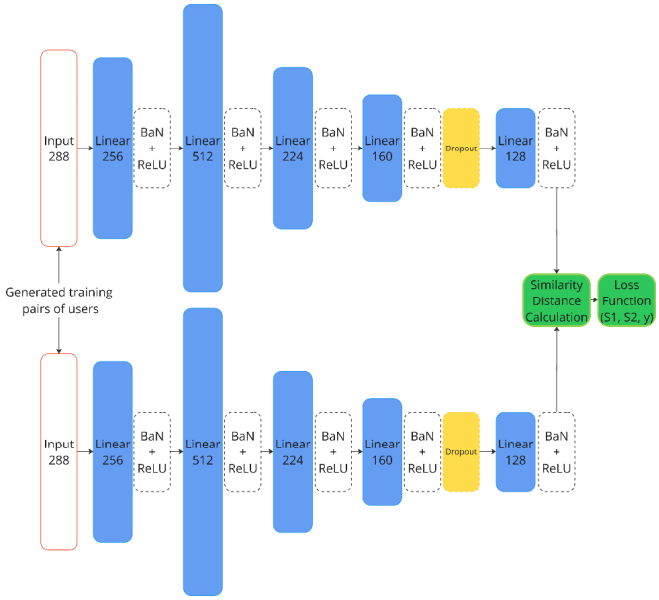

Fig. 2. Configured structure of siamese networks.

IV. RESULTS

To validate the hypothesis about the effectiveness of autoencoders in generating dense and compressed embedding vector representations, a comparison with PCA was carried out, a widely used traditional dimensionality reduction technique [24]. PCA applies a linear transformation, projecting data onto a lower-dimensional space by maximizing variance along principal components. However, it lacks the ability to capture nonlinear dependencies, which are crucial for modeling complex relationships in user behavior [25]. Unlike PCA, autoencoders utilize nonlinear activation functions, such as ReLU and LeakyReLU, allowing them to learn hierarchical and nonlinear representations of the data. This enables them to preserve intricate patterns and dependencies that cannot be captured by linear techniques.

Figure 3 presents a comparison between the two dimensionality-reduction methods applied to the same user dataset. Both images depict a 3D projection of the original high-dimensional feature space. The PCA projection reveals a sparse and dispersed distribution of data points, reflecting the limitations of linear transformations in capturing the structure of complex behavioral data. In contrast, the autoencoder forms a denser and more coherent cluster, indicating a better preservation of semantic relationships within the data [26]. This visualization supports the argument that deep learning-based embeddings are more effective than classical linear methods in modeling nonlinear dependencies in high-dimensional user data and highlights the practical benefits of using neural network-based compression techniques for similarity-based user modeling.

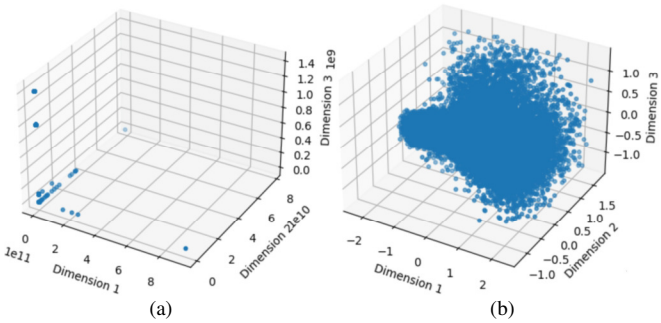

Fig. 3. Experimental modular architecture: (a) PCA, (b) autoencoder-based embedding.

To further assess the practical advantages of autoencoder-generated embeddings, the positioning of data points within the n-dimensional latent space was examined to determine whether it exhibited meaningful structural properties. Additionally, the ability to effectively cluster observations was evaluated, ensuring that similar instances were grouped closely together while dissimilar ones were positioned farther apart. These observations further validate the model's effectiveness in look-alike search applications.

To support this analysis, the data was visualized with respect to the operating systems of the devices. Figure 4 visualizes the distribution of user devices in the latent embedding space. The left subplot illustrates clustering patterns based on the device OS. Devices running iOS are shown in pink, while Android devices appear in shades of purple. The broader dispersion among Android devices reflects the diversity of manufacturers and the wide range of device characteristics and price segments. The right subplot highlights iOS devices in the embedding space. These devices form compact and well-defined clusters, suggesting that the model effectively captures underlying similarities between user profiles associated with Apple products. This clustering supports the claim that the embedding space preserves semantically meaningful relationships and is suitable for user segmentation and look-alike modeling tasks.

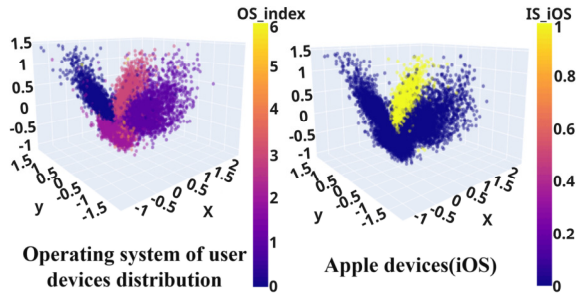

Fig. 4. Data visualization with a categorical breakdown.

This analysis demonstrates that the embedding vectors exhibit distinct physical properties, as evidenced by their ability to represent meaningful relationships and form coherent clusters within the data. These findings further validate the hypothesis, showing that the embedding space effectively captures and reflects key characteristics of the input data.

Following the training of the model on the training dataset, its performance was validated on the test dataset. As noted previously, Siamese networks were employed to measure similarity between vector representations, with Euclidean distance selected as the primary metric due to its simplicity and effectiveness in high-dimensional embedding spaces.

Based on the findings of this study, both cosine similarity and siamese networks can be applied for similar audience searches, as they demonstrated superior performance in different aspects of look-alike audience identification. Cosine similarity is particularly effective in cases where relative feature relationships are more significant than absolute distances. This method ensures high accuracy in high-dimensional spaces by focusing on directional similarity rather than magnitude, making it robust to variations in scale across different user profiles. Its computational efficiency also allows for rapid similarity calculations, making it a scalable solution for large datasets. On the other hand, Siamese networks offer a learned similarity function, adapting to the specific characteristics of the dataset instead of relying on predefined distance measures. This approach captures complex, nonlinear relationships between users, improving the overall quality of look-alike audience selection by learning task-specific feature representations. While Siamese networks require higher computational costs during training, they provide better discrimination between similar and dissimilar users once embeddings are precomputed, making them ideal for applications requiring high precision and adaptive similarity learning. Given these results, the choice between cosine similarity and Siamese networks depends on the specific use case, as the former is preferable when computational efficiency and scalability are prioritized, while the latter excels in applications requiring higher accuracy and adaptive learning.

A comprehensive evaluation of the model's performance was carried out using metrics such as lift, precision, recall, and F1 score. The Lift top 1 metric evaluates the model's ability to outperform a baseline by focusing on the top 1% of predictions ranked by score, providing insights into its prioritization accuracy. This metric offers insights into the model's ability to outperform a baseline in identifying high-priority predictions.

In addition to these metrics, a business-oriented performance measure, the conversion metric, was incorporated to assess the model's impact on real-world outcomes. This metric evaluates the model's ability to convert predictions into desired actions, such as user engagement or responses to targeted advertisements.

The Precision-Recall (PR) curve played a crucial role in determining the optimal decision threshold for targeted advertising. By analyzing the PR curve, the threshold was fine-tuned to achieve an optimal balance between precision and recall, ensuring that the model delivers both high targeting accuracy and sufficient audience reach. The choice of threshold significantly impacts the practical application of the model. A higher threshold improves precision, selecting only the most relevant users and minimizing marketing budget waste, but it may reduce the overall audience size. Conversely, a lower threshold increases recall, capturing a broader audience but potentially increasing false positives, which could lead to lower engagement rates and higher customer acquisition costs. To optimize this trade-off, the threshold was adjusted based on historical campaign performance, allowing flexible adaptation to different advertising goals. This ensures that B2B clients can prioritize highly qualified leads or expand their reach according to their strategic objectives, maximizing the overall effectiveness of their campaigns.

As shown by the results in Table I, the use of Siamese networks in combination with autoencoders not only significantly accelerated the model's performance but also enabled the discovery of hidden nonlinear dependencies within the data. This approach improved the model's results across all evaluated metrics, solidifying its effectiveness in the task.

TABLE I. TEST RESULTS

Model Type	Metric value					
	Lift-Top1	**ROC-AUC**	**Recall**	**Precision**	**F1 score**	**CR**
SVM no entities embeddings [1]	4.9	0.64	0.55	0.54	0.54	0.13
RF - no entities embeddings [1]	5.4	0.66	0.54	0.60	0.57	0.15
LightGBM - no entities embeddings [1]	6.6	0.69	0.56	0.64	0.60	0.19
Cosine similarity with all concatenated entity embeddings	11.7	0.76	0.70	0.73	0.71	0.31
Siamese networks with all concatenated entity embeddings	12.9	0.79	0.74	0.75	0.74	0.36

From a business perspective, these improvements directly affect the efficiency of advertising campaigns and operational cost savings. A higher lift score indicates that the model identifies users with a higher likelihood of engagement, leading to higher conversion rates and better return on investment for advertisers. Increased precision and recall ensure that fewer irrelevant users are targeted, reducing wasted ad impressions and optimizing marketing budgets. For advertising departments, this results in lower customer acquisition costs, improved targeting efficiency, and a more scalable advertising strategy. Furthermore, since the model is integrated into an automated service, no additional human resources, such as business development managers or data analysts, are required for manual processing. This automation allows the service provider to save on human resource costs while enabling faster request handling for B2B clients. As a result, B2B customers receive quick and expert-level analytics from the telecommunications provider, significantly accelerating decision-making and enhancing the efficiency of their advertising strategies. Using the improved model, businesses can allocate resources more effectively, ultimately achieving greater profitability and marketing effectiveness while benefiting from fully automated audience selection and analysis. The proposed model is designed to be scalable and adaptable for large-scale industrial applications.

The use of precomputed embeddings significantly reduces the computational cost during inference, allowing real-time similarity calculations in look-alike audience searches. In terms

of deployment feasibility, Siamese networks require higher computational resources during training due to the need for processing paired inputs, making initial model training more time-intensive compared to traditional methods such as SVM or RF. However, once trained, the model can efficiently process new user data by generating embeddings and performing similarity calculations with minimal computational overhead.

For large-scale implementation, the model can be integrated into cloud-based infrastructures with GPU acceleration, ensuring fast processing times for massive user databases. Additionally, the approach can be containerized and deployed as a microservice, allowing seamless integration into existing advertising platforms. If a company already operates automated data pipelines that generate user profiles for analytics or recommendation systems, it can implement a similar model by applying dimensionality reduction techniques to create dense user embeddings. These embeddings can then be processed using any similarity scoring algorithm explored in this study, depending on business needs.

The training and automated scoring processes can be executed on the same computational resources used for existing user profile calculations. However, if more advanced similarity search methods, such as Siamese networks, are utilized, additional GPU-powered infrastructure may be required to handle the increased computational complexity efficiently. Although the training phase requires substantial computational resources, the overall operational cost remains manageable, given that inference is efficient and suitable for real-time applications. This makes the model practical and scalable for industrial use cases in targeted advertising and telecommunications.

V. CONCLUSION

This study presented a thorough analysis of autoencoder architectures and their ability to address the curse of dimensionality. It emphasized the role of autoencoders in enhancing the robustness of machine learning models, particularly when integrated with Siamese networks for handling high-dimensional data. Compared to traditional methods for tabular data and classical machine learning approaches, this combination exhibits superior performance, scalability, and adaptability. Empirical results confirmed that the proposed approach outperforms SVM, RF, and LightGBM in all major evaluation metrics, including lift, ROC AUC, recall, precision, and F1 score. In particular, the Siamese network with concatenated embeddings achieved a Lift-Top 1 of 12.9 and a conversion rate of 0.36, significantly exceeding all baseline models. These results highlight the method's practical advantages in look-alike audience identification and its effectiveness in high-dimensional advertising scenarios.

The novelty of this research lies not only in the innovative application of Siamese networks and autoencoders to tackle tabular data challenges in telecommunications but also in the complex approach adopted. This includes the development of a flexible look-alike model by combining embedding vectors derived from multiple entities, thereby enhancing the model's adaptability and generalization capabilities. By leveraging autoencoders to generate dense and compressed embedding vectors, this study provides an effective solution to the curse of dimensionality, enhancing the representation of complex patterns and nonlinear dependencies. Compared to traditional methods, autoencoders have demonstrated strong capabilities in constructing a universal look-alike search model, offering significant improvements in computational efficiency, accuracy, and feature diversity [1]. Furthermore, this research investigates the structural properties of compressed embedding spaces, highlighting their potential for broader applications beyond targeted advertising, particularly in domains where high-dimensional data representation plays a crucial role.

The fusion of embedding vectors introduces a novel approach to integrating information from diverse entities. Unlike traditional methods that treat each entity in isolation, this approach consolidates compressed representations of users, including their web and financial activity, devices, base stations, and tariff plans, into a single vector. This integration enables the creation of a comprehensive user profile, significantly enhancing the model's capability to uncover intricate relationships and dependencies across multiple domains. Moreover, the incorporation of Siamese networks as a metric for evaluating vector similarity marks a substantial advancement. This approach enables precise measurement of the similarity between compressed embedding vectors, utilizing distances calculated by the Siamese network. The study proposes this sophisticated method to compare user representations, allowing a more granular evaluation of similarities and differences. This approach not only surpasses traditional methods but also introduces a new level of complexity, allowing the model to more effectively identify users with similar characteristics.

This research also tackled the challenge of limited negative examples in data using autoencoders to develop look-alike models in the telecommunications domain. The primary objective was to predict subscriber interest in advertisements using historical data and behavioral patterns. To achieve this, embedding vector representations were created for six distinct entity domains. The autoencoder was trained to compress the data into dense representations, capturing essential features. The Siamese network was then used to compute the similarity between pairs of concatenated embeddings, with a predefined cutoff threshold identifying whether a subscriber shared behavioral similarities with users who had previously engaged with the target action. The results demonstrated strong performance across various evaluation metrics, confirming the reliability of the proposed approach. The proposed approach demonstrates versatility, with potential applications in finance for customer behavior analysis and in healthcare for predictive modeling using patient records. In finance, for instance, the same framework could analyze transaction patterns to uncover trends in customer behavior. In healthcare, it could be used to compress and model extensive tabular datasets, such as patient diagnostic records, for predictive purposes. By revealing hidden nonlinear dependencies and minimizing information loss, this approach offers an efficient solution for analyzing large-scale tabular datasets. Its adaptability positions it as a valuable tool across multiple industries to extract insights and enable data-driven decision-making.

Additionally, integrating a Siamese network into the model architecture broadens its applicability to tasks such as anomaly detection. For example, this method can detect unusual subscriber behavior, helping to identify service disruptions more quickly and improve overall quality. Furthermore, the framework can support the development of recommendation systems by analyzing subscriber behavior patterns, leading to more personalized and effective services.

Future research will explore alternative similarity metrics for embedding comparisons and expand the model's feature set by integrating additional entity domains. These enhancements will allow the model to incorporate additional dimensions of knowledge, thus improving its robustness and expanding its application potential.

REFERENCES

[1] A. Altaibek, I. Tokhtakhunov, M. Nurtas, D. Kozhamzharova, and M. Aitimov, "The Efficacy of Autoencoders in the Utilization of Tabular Data for Classification Tasks," *Procedia Computer Science*, vol. 238, pp. 492–502, 2024, https://doi.org/10.1016/j.procs.2024.06.052.

[2] K. Mamta and S. Sangwan, "AaPiDL: an ensemble deep learning-based predictive framework for analyzing customer behaviour and enhancing sales in e-commerce systems," *International Journal of Information Technology*, vol. 16, no. 5, pp. 3019–3025, Jun. 2024, https://doi.org/10.1007/s41870-024-01796-z.

[3] R. Gustriansyah, J. Alie, and N. Suhandi, "A Hybrid Machine Learning Model for Market Clustering," *Engineering, Technology & Applied Science Research*, vol. 14, no. 6, pp. 18824–18828, Dec. 2024, https://doi.org/10.48084/etasr.9259.

[4] W. Wang, "Dimensionality Reduction Task," in *Principles of Machine Learning*, Springer Nature Singapore, 2025, pp. 481–505.

[5] S. A. Wegner, "Curse and Blessing of High Dimensionality," in *Mathematical Introduction to Data Science*, Springer Berlin Heidelberg, 2024, pp. 115–125.

[6] Y. Bengio, L. Yao, G. Alain, and P. Vincent, "Generalized Denoising Auto-Encoders as Generative Models," in *Advances in Neural Information Processing Systems*, 2013, vol. 26, [Online]. Available: https://proceedings.neurips.cc/paper/2013/hash/559cb990c9dffd8675f6bc2186971dc2-Abstract.html.

[7] H. S. Lom, A. C. Thoo, W. M. Lim, and K. Y. Koay, "Advertising value and privacy concerns in mobile advertising: the case of SMS advertising in banking," *Journal of Financial Services Marketing*, vol. 29, no. 3, pp. 1135–1153, Sep. 2024, https://doi.org/10.1057/s41264-023-00263-3.

[8] N. Capuano, M. Meyer, and F. D. Nota, "Analyzing the impact of conversation structure on predicting persuasive comments online," *Journal of Ambient Intelligence and Humanized Computing*, vol. 15, no. 11, pp. 3719–3732, Nov. 2024, https://doi.org/10.1007/s12652-024-04841-8.

[9] S. Merugu, R. Yadav, V. Pathi, and H. R. Perianayagam, "Identification and Improvement of Image Similarity using Autoencoder," *Engineering, Technology & Applied Science Research*, vol. 14, no. 4, pp. 15541–15546, Aug. 2024, https://doi.org/10.48084/etasr.7548.

[10] W. Lee, S. Lee, H. Kim, and J. Lee, "Sliced Wasserstein adversarial training for improving adversarial robustness," *Journal of Ambient Intelligence and Humanized Computing*, vol. 15, no. 8, pp. 3229–3242, Aug. 2024, https://doi.org/10.1007/s12652-024-04791-1.

[11] M. A. Javed *et al.*, "Leveraging Convolutional Neural Network (CNN)-based Auto Encoders for Enhanced Anomaly Detection in High-Dimensional Datasets," *Engineering, Technology & Applied Science Research*, vol. 14, no. 6, pp. 17894–17899, Dec. 2024, https://doi.org/10.48084/etasr.8619.

[12] K. M. Ghori, M. Imran, A. Nawaz, R. A. Abbasi, A. Ullah, and L. Szathmary, "Performance analysis of machine learning classifiers for non-technical loss detection," *Journal of Ambient Intelligence and Humanized Computing*, vol. 14, no. 11, pp. 15327–15342, Nov. 2023, https://doi.org/10.1007/s12652-019-01649-9.

[13] P. Vincent, H. Larochelle, Y. Bengio, and P. A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proceedings of the 25th international conference on machine learning - ICML '08*, Helsinki, Finland, 2008, pp. 1096–1103, https://doi.org/10.1145/1390156.1390294.

[14] Z. Hu, Z. Xiao, H. Sun, and H. Yang, "Autoencoder evolutionary algorithm for large-scale multi-objective optimization problem," *International Journal of Machine Learning and Cybernetics*, vol. 15, no. 11, pp. 5159–5172, Nov. 2024, https://doi.org/10.1007/s13042-024-02221-4.

[15] T. O. Hodson, "Root-mean-square error (RMSE) or mean absolute error (MAE): when to use them or not," *Geoscientific Model Development*, vol. 15, no. 14, pp. 5481–5487, Jul. 2022, https://doi.org/10.5194/gmd-15-5481-2022.

[16] N. Serrano and A. Bellogín, "Siamese neural networks in recommendation," *Neural Computing and Applications*, vol. 35, no. 19, pp. 13941–13953, Jul. 2023, https://doi.org/10.1007/s00521-023-08610-0.

[17] F. Baier, S. Mair, and S. G. Fadel, "Self-supervised Siamese Autoencoders," in *Advances in Intelligent Data Analysis XXII*, vol. 14641, I. Miliou, N. Piatkowski, and P. Papapetrou, Eds. Cham: Springer Nature Switzerland, 2024, pp. 117–128.

[18] Y. Zhang *et al.*, "Similarity-based pairing improves efficiency of siamese neural networks for regression tasks and uncertainty quantification," *Journal of Cheminformatics*, vol. 15, no. 1, Aug. 2023, Art. no. 75, https://doi.org/10.1186/s13321-023-00744-6.

[19] A. Fedele, R. Guidotti, and D. Pedreschi, "Explaining Siamese networks in few-shot learning," *Machine Learning*, vol. 113, no. 10, pp. 7723–7760, Oct. 2024, https://doi.org/10.1007/s10994-024-06529-8.

[20] W. Q. Yan, "Generative Adversarial Networks and Siamese Nets," in *Computational Methods for Deep Learning*, Springer Nature Singapore, 2023, pp. 125–140.

[21] A. J. Chemmanam, B. Jose, and A. Moopan, "Improved multi object tracking with locality sensitive hashing," *Pattern Analysis and Applications*, vol. 27, no. 4, Dec. 2024, Art. no. 136, https://doi.org/10.1007/s10044-024-01353-1.

[22] Q. Ma, M. Wen, Z. Xia, and D. Chen, "A Sub-linear, Massive-scale Look-alike Audience Extension System A Massive-scale Look-alike Audience Extension," in *Proceedings of the 5th International Workshop on Big Data, Streams and Heterogeneous Source Mining: Algorithms, Systems, Programming Models and Applications at KDD 2016*, Dec. 2016, pp. 51–67. [Online]. Available: https://proceedings.mlr.press/v53/ma16.html.

[23] O. Rainio, J. Teuho, and R. Klén, "Evaluation metrics and statistical tests for machine learning," *Scientific Reports*, vol. 14, no. 1, Mar. 2024, Art. no. 6086, https://doi.org/10.1038/s41598-024-56706-x.

[24] K. Berahmand, F. Daneshfar, E. S. Salehi, Y. Li, and Y. Xu, "Autoencoders and their applications in machine learning: a survey," *Artificial Intelligence Review*, vol. 57, no. 2, Feb. 2024, Art. no. 28, https://doi.org/10.1007/s10462-023-10662-6.

[25] M. Nurtas, *et al.*, "Predicting the Likelihood of an Earthquake by Leveraging Volumetric Statistical Data through Machine Learning Techniques," *Engineered Science*, 2023, https://doi.org/10.30919/es1031.

[26] M. Nurtas, Z. Zhantaev, and A. Altaibek, "Earthquake time-series forecast in Kazakhstan territory: Forecasting accuracy with SARIMAX," *Procedia Computer Science*, vol. 231, pp. 353–358, 2024, https://doi.org/10.1016/j.procs.2023.12.216.