Future Generation Computer Systems 127 (2022) 70–79

Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

Reliable customer analysis using federated learning and exploring deep-attention edge intelligence

Usman Ahmed [a], Gautam Srivastava [b,c], Jerry Chun-Wei Lin [a,*]

[a] Department of Computer Science, Electrical Engineering and Mathematical Sciences, Western Norway University of Applied Sciences, 5063, Bergen, Norway
[b] Department of Math and Computer Science, Brandon University, Brandon, Canada
[c] Research Centre for Interneural Computing, China Medical University, Taichung, Taiwan

ARTICLE INFO

Article history:
Received 6 February 2021
Received in revised form 29 June 2021
Accepted 28 August 2021
Available online 2 September 2021

Keywords:
Federated learning
Edge intelligence
Mobile wireless networks
Attention network
Clustering
Market analysis

ABSTRACT

The Internet of Things (IoT) and smart cities are flourishing with distributed systems in mobile wireless networks. As a result, an enormous amount of data are being generated for devices at the network edge. This results in privacy concerns, sensor data management issues, and data utilization issues. In this research, we propose a collaborative clustering method where the exchange of raw data is not required. The attention-based model used with a federated learning framework. The edge devices compute the model updates using local data and send them to the server for aggregation. Repetition is performed in multiple rounds until a convergence point reached. The transaction data used to train the attention model that gives a low dimensional embedding. Afterwards, we share the convergence model among the client/stores. Then, efficient pattern mining methods known as a clustering-based dynamic method (CBDM) are applied. For experimentation, we used retail store data to cluster the customer based on purchase behaviour. The proposed clustering method used semantic embedding to extract and then cluster them by discovering relevant patterns. The method achieved the 0.75 ROC values for the random distribution and 0.70 for the fixed distribution. The clustering method can help to reduce communication costs while ensuring privacy.

1. Introduction

Pattern analysis methods tend to have a different scope of application in distributed systems. The core purpose of pattern analysis is to find structurally similar patterns in a database. Retail companies use customer purchase behaviour to update stocks as a well-made critical decision mechanism to get an edge against competitors [1]. Enterprises use modelling methods to learn the behaviour of customers [2]. For example, customer buying patterns of certain services or products, preference regarding prices and time, and the impact of a decision made because of analysis. This analysis helps retail companies serve the customers better and, as a result, increase profits for them. Maintaining the dynamic nature of the market retails companies/enterprises required advanced technologies and always in the hunt for the data. This gives rise to an additional question, i.e., is it possible to combine data in a common site by transporting data across the organization using different databases? It is very difficult to maintain the database of the number of sales points, if not impossible. Furthermore, it is common to see mobile wireless Point of Sales devices used in stores where real-time data used to complete sales transactions. Many situations result in barriers to the sharing of data sources. Data required by the pattern analysis methods involve multiples types. For instance, in a product recommendation system service, the enterprises or owners have the records (product information and data of the user's purchase) but not the data that describes the payment habits and purchasing ability. In most industries, databases extant in the form of confined islands. This is because of market competition, customer privacy, complicated administrative procedures, and even resistance in intra-department data integration. For this reason, it becomes impossible to integrate scattered data across departments around the country and institutions or because it prohibits the cost.

With the increasing number of incidents related to data and user privacy, companies emphasize data security issues. Data leakage damages the reputation of companies and raises concerns within public opinion and the government as well. For example, the recent data breach in Facebook resulted in a wide range of protests [3]. As a result, different states in the United States of America (USA) reevaluated their data protection laws. The prime example of data enforcement is the introduction and implementation of the General Data Protection Regulation (GDPR) by the

* Corresponding author.
E-mail addresses: usman.ahmed@hvl.no (U. Ahmed), jerrylin@ieee.org (J.C.-W. Lin).

https://doi.org/10.1016/j.future.2021.08.028

European Union on May 25, 2018.[1] The law aims to give legal protection for the user and grants the "right to be forgotten", that is, users can have their data deleted or withdrawn. Heavy fines will impose if companies found violating the bill. The United States and China also took similar action. The China Cybersecurity Law and the General Principles of Civil law (2017) are required to secure data. The law also enforces the contact (legal binding and protection) to avoid leakage issues when conducting data transactions with the third party. They make the legal data protection obligations a requirement for the contract. These legal binding and establishment of new regulations help to build a secure data society. However, a new challenge arises for applying the common pattern analysis method to the data without sharing it.

Specifically, pattern analysis models often involve transaction models that included steps, i.e., collection, transferring data to another party (server), the cleaning, and the party to implement fusing requirements [2]. Then, the clean data used by the third-party to integrate, build and extract pattern helps to make a practical model (depending on the application) for making useful decisions. They usually sell the model as a service for different enterprises. Now, the legal binding of data protection laws results in challenges with the above new data regulations and laws. As users do not know the future usage of the models, the traditional methods violate laws such as the GDPR. Therefore, the collected data results in an isolated island where the analyst cannot collect, integrate, analyse, and use it for decision-making.

1.1. Motivation

Solving data fragmentation (right data, in the right format and quantity) is challenging for practitioners today. We propose one solution for the above problem in this research, i.e., reliable customer analysis, by exploring an attention-based federated learning model useable at the edge of mobile wireless networks and in conjunction with standard wired architecture [4,5]. We attempt to demonstrate how customer clustering can be implemented. Learned knowledge can be shared using a federated learning (multiple nodes connected by a distributed channel) approach at the edge. The database is split into similar clusters, and analysis will reduce the processing cost of the existing pattern mining algorithms. Clusters could potentially occur on a collection of mobile devices or similar collections of devices capable of handling the required computational load.

Market analysis requires transaction data, which is made up of a set of distinct items/products. The item/product consumption represents distinct patterns according to distinct behaviour [6]. The patterns have different orientations (data scalability, modality, convexity, and correlation) and often require different actions. The most common method for pattern analysis is using the frequency of products purchased and used by customers [7]. A bitmap is a value-based representation of vector representation. If the item exists, then a bit/value is assigned to it. Otherwise, it is set to zero. The frequent item (sets) are represented within a huge vector space since the number of items (sets) is often huge, especially when data are collected from distributed environments. This often results in the curse of high dimensionality and data sparsity problems. However, the methods have four limitations.

1. First, the model is particularly built for specific tasks, i.e., utility mining, sequence prediction, or classification. As a result, market pattern analysis is only applied to custom data. The analysis/learning cannot be transfer or applied to other applications or markets.
2. Second, the methods required the structured database. However, the real-world applications are often dynamics and do not allow multiple scans of the database, i.e., stream data.
3. Third, the model required a scalable amount of data to extract patterns. The databases often collaborate in the centralized structure. This structuring often results in overhead as it requires time-consuming because of the above issues (data privacy and ethical concerns). Even when the challenges being addressed, some organizations considered the data to be more valuable than the application, so they do not prefer to share the full data sets [8].
4. Fourth, the data set of distributed data are often very large and expensive to gain the storage for central hosting. For this reason, the federated learning approach [9] can tackle the above issues, where only model weights are shared across the network without the raw data information.

1.2. Contributions

We propose an attention-based federated learning approach for customer analysis at the edge of mobile networks. For transaction data, the model learns low dimensional representation (embedding) in a fully unsupervised manner. We use the utility [2] *(quantity* $\times$ *profit)* of the item as the input vector and then reduce it to a low dimensional space using the attention-based model. The locally trained model and sharing the model weights by using federated learning helps to increase the performance globally and locally. The low-dimensional embedding helps capture the product and customer relations as a result and detect dynamic communities. After that, the embedding is used by the proposed dynamic clustering model that is used to build clusters. Thus, developing attention federated learning and dynamic clustering helps capture the relationships among the transactions for a real-world edge intelligence application. In short, the paper contributions are as follows:

1. Propose privacy preserved utility transaction embedding model for customer analysis that uses the positional attention model for distributed stores at the edge.
2. Propose a dynamic clustering model that takes the low dimension semantics aware embedding to give meaningful clusters.
3. Investigate the impact of applying federated learning and dynamic clustering on real market store data at the edge of a given mobile network. The learned embedding helps to improve the performances of dynamic clustering without raw data sharing, and raw data can remain on mobile edge devices without being openly transmitted.

2. Related work

Over the past few decades, pattern mining algorithms [2] showed the effectiveness to discover valuable information for decision-making. The databases are diverging, such as binary, quantitative, probabilistic, and fuzzy databases. Mining information from them is a non-trivial process that requires contemplation of various significant constraints [2,10]. Generally, mining algorithms are different from each other based on the data structure and pruning strategy. The purpose of mining algorithms is to find useful patterns that can be utilized in real-world applications. Mining algorithms are often used for determining the usefulness of patterns for research purposes and often make use of different types of constraints, i.e. support, confidence and utility parameters (e.g., price, profit, weight, uncertainty, quantity, satisfaction, etc.) [1,2,10]. This surge results in demand utility-oriented patterns mining that have high impact application in the market analysis, e-commerce, finance and medical.

[1] https://op.europa.eu/en/publication-detail/-/publication/3e485e15-11bd-11e6-ba9a-01aa75ed71a1/.

2.1. Clustering of heterogeneous databases

In the study, Fuzzy C-Means (FCM) and *k*-means algorithms were proposed to identify the relevant sources of information and discard the relevant ones using their meta-data information [11]. Different forensic case series were employed to assess the performance of algorithms. The clustering helps to identify the related sources. All the candidate clusters grouped into one by using different configuration proportions.

In the study, [12], a stock exchange prediction model was proposed to combine information from two corporations. A graph of all the corporations was formed using investment facts information, and the node embedding method was applied to learn distributed representation of each paragraph. A pipeline model and joint model-based approaches were explored to harness information of associated corporations. The experimental results were evaluated on the stock market data set. The model successfully captured relations among corporations.

Similarly, *k*-means clustering-based framework [13] was proposed to mine information from heterogeneous data that contain data from multiple sources. The authors argued that simple clustering methods understand the values of only homogeneous attributes. Whereas in real-life scenarios, data are vastly heterogeneous. Therefore, they applied the *k*-means clustering algorithm on real-life heterogeneous data sets, and all the generated clustered were assessed rigorously.

The unsupervised clustering method is also proposed [14] that initializes without any parameter selection and finds an optimal number of clusters. The method used entropy-type penalty terms to construct the schema. The approach utilizes the structure of the data and then initialized the number of clusters. Evolutionary computation-based clustering techniques is also proposed that selects the density of the clusters and evaluate it by using the fitness criterion [15]. Then, iterative searches for the globally optimized solutions. The model uses the minimum distance principle to reduce the centre points. As result closest point results as the clusters.

2.2. Deep neural networks

Deep learning has become an emerging topic in recent years since it can achieve better performance, i.e., accuracy than the past generic machine learning models, which has been applied in many domains and applications [16,17]. Hidden patterns and high dimension features often help the neural network learn the distinct representation of feature space [18]. The trained network then uses the learned features to the computer's conditional distribution of input vectors. The different architecture of the neural network is being proposed for domain-specific applications. One of the basic is that the architecture is multi-layer perceptron [18]. Each hidden layer takes averaging layers of outputs in this network to compute input from the previous layer and weights. The non-linear activation function is used at the final/output layer of the network. They update the weights based on the loss function and gradient.

In supervised learning, the network reduces the loss and is considered a non-linear optimization problem. The weight and bias values used to optimize the loss. The algorithms mostly fall under the gradient descent technique. The gradient-based techniques start with random points for each input vector. It then several iterations (epochs) are executed for a set of the instance (batches). The trainer computes the loss; it was made by computing the non-linear objective function for the loss values and gradient. Then, weights updated in a way that reduces the loss function [18]. The loss is continuously reduced to the convergence point or optimal local minimum. The predictive ability of the neural networks comes from hidden layers and the structure of the architecture. The correct selection of several layers, architecture structure, layers, and hyperparameters helps solve complex problems. The higher-order representation of the input features vector is achieved using the network training [19]. The learned higher feature representation helps to achieve generalization and increase predictive power. Modern research in the neural network selects the network with low computation complexity and has high prediction power. The number of architecture is proposed over the past two decades [20].

The major difference between architectures is the hidden layers, layers type, shapes, and connection between layers [21]. Wainberg et al. introduced the method to learn higher-dimensional features from the tabular data [22]. The convolutional neural network (CNN) learns features embedding from the image pixels. The pixel data and variation among them increase the learning and predictive power of the network. The translation invariant pixel benefits the network [22]. Many studies were conducted on learning and inference in the visual information processing system that includes wildlife application [23], X-ray scans [24], and autonomous driving [20]. For sequential data, recurrent neural network (RNN) architecture was proposed and used in the domain of natural language process includes machine translation, language generation, and time series analysis [25–27]. The RNN model comprises an encoder and decoder framework where the encoder takes the input sequence and decodes it into the vector's fixed length. The model uses different gates to process the input features based on the loss function. The fixed-length vector sometimes loses relevant information [19].

Another issue with the RNN encoder and decoder model is the alignment of the input and output vector. Neighbour feature values influence the sequence. Another variant of RNN is the proposal of a new network named as attention mechanism [19]. It applies the attention method to the input vector by giving certain weights to selected inputs. This selection based on the prioritized importance and position of relevant information. After that decoder used the position with context vector and corresponding weights for the higher feature representation. After that mode is then learned the weights to the RNN model for the predictions [19], the attention weights and context vector learned by using the architecture and feature representation [28]. Several variations of the network include a soft, hard, and global architecture for the attention mechanism. They proposed the soft attention model [29] to help reduce contextual information. The model used the average of the hidden states and then built the context vector. The approach helps to efficiently learn the input feature hidden pattern and reduce the loss.

In hard attention, Xu et al. computes the context vector from sampling the hidden states [30]. The hard attention reduces the computation cost; however, tuning the architecture is difficult as the convergence of architecture is difficult. Luong et al. [31] proposes another variation, i.e.,local and global attention. Global attention is the intermediate version of soft and hard attention. The model picks the attention point for each input batch. This helps to reach convergence quickly. In the local attention model, they learn the position of the attention vector from the predictive function. The model predicts the attention position. Both local and global attention computationally efficient and requires to be selected by analysing the domain-specific data.

2.3. Federated learning

Federated learning can be classified into two distinct frameworks. i.e., horizontal and vertical federated learning [32]. In horizontal federated, data distribution among nodes is different; however, feature space is the same [33]. The method has overlapping characteristics with privacy preservation machine learning

Table 1
The transaction quantitative database.

T_{id}	Item: quantity
T_1	(a:5); (b:3); (c:6)
T_2	(c:4); (d:2)
T_3	(a:7); (b:8); (e:2)
T_4	(a:3); (c:1)
T_5	(b:2); (c:4); (e:4)

Table 2
The unit profit of the items.

Item	Profit
a	8
b	3
c	8
d	3
e	5

as it considers the decentralized collaborative learning [32]. In vertical federated learning, the dataset features are different; however, data distribution is overlapping [32]. The mechanism of federated learning is proposed by Shokri et al. [33]. They train the model on multiple learning techniques on joint inputs. Hayes and Ohrimenko [34] proposed the model sharing through the trusted mechanism. The using federated learning output and optimization method is proposed by Fedrikson et al. [35]. Mohassel and Rindal [36] proposed the aggregation function that uses the approximation of fixed-point multiplication protocols.

In general, data mining and prediction models (machine learning and deep learning) are used for prediction and clustering problems. However, preventing data leakage during pre-processing and communication remains a challenging task. To prevent data leaks when using a distributed dataset, federated learning (FL) methods have been proposed to be used at the edge of networks. These FL frameworks only share model weights (black boxes) to network nodes, and the model trains local nodes (client) using the actual dataset. The problem at hand can be solved by combining federated learning, attention mechanism, and dynamic clustering. The issues present include data distribution, low instances, device ability for optimization.

Machine learning over distributed data stored by many clients has important applications. Where data privacy is a key concern, or central data storage is not an option. Hegeds et al. proposed the master–worker architecture, the workers do machine learning over their data, and the master aggregates the resulting models without seeing any raw data [37]. The method proposed *Gossip learning* a decentralized alternative to federated learning that does not require an aggregation server or any central component. The experimental scenarios include traces collected over mobile phones and continuous communication patterns with different network sizes and distributions of the training data over the devices. The aggregated cost of machine learning in both approaches is also examined. The best gossip variants do comparably to the best-federated learning variants, offer a fully decentralized alternative to federated learning.

Distributed devices often produce a large amount of data, eventually resulting in big data that can be vital in uncovering hidden patterns and other insights in numerous fields such as healthcare, banking, and policing. Chamikara et al. [38] shows different attack methods such as membership inference that exploit the vulnerabilities of ML models and the coordinating servers to retrieve private data. In addition, big data often requires more resources than available in a standard computer, and Chamikara et al. also proposed a distributed perturbation algorithm named DISTPAB for privacy preservation of horizontally partitioned data. DISTPAB alleviates computational bottlenecks by distributing the task of privacy preservation utilizing the asymmetry of resources of a distributed environment, which can have resource-constrained devices and high-performance computers.

3. Methodology

Let us consider $I = \{i_1, i_2, \ldots, i_m\}$ as a set of items, and a set of transactions from databases are $D = \{T_1, T_2, \ldots, T_n\}$. Examples of quantitative databases are illustrated in Table 1. The items' unit profit in the database is also shown in Table 2.

Table 3
Transaction-itemset-selection as input features.

T_{ID}	a	b	c	d	e	Utility
T_1	40	9	48	0	0	$97
T_2	0	0	24	6	0	$38
T_3	56	24	0	0	10	$90
T_4	24	0	6	0	0	$32
T_5	0	6	24	0	20	$58

Definition 3.1 (*Item Utility in a Transaction*)**.** The utility of an item i_k is $u(i_k, T_c)$ in T_c, the transaction, defined as follows:

$$u(i_k, T_c) = pr(i_k) \times q(i_k, T_c). \tag{1}$$

Example 3.1. For example, the calculation of utility for an item (a) in T_1 is as follows: $u(a, T_1) = 5 \times 8 = \$40$.

Definition 3.2 (*Itemset Utility in a Database*)**.** In the database transaction D, the utility of X is defined as $u(X)$:

$$u(X) = \sum_{X \subseteq T_c \wedge T_c \epsilon D} u(X, T_c). \tag{2}$$

Example 3.2. For instance, the utility of an itemset (ac) in T_1 is computed as: $u(ac) = u(a, T_1) + u(c, T_1) + u(a, T_4) + u(c, T_4) = \$40 + \$48 + \$24 + \$8 = \$120$.

Consider Tables 1 and 2 as an example that will be executed according to the proposed algorithm. There are five transactions in the table (T_1, T_2, T_3, T_4, T_5). For instance, transaction T_2 exhibits the items (c) and (d); and quantities of their purchase which are 4 and 2 respectively. The unit of profit earned against each sold item of the transaction is shown in Table 2, i.e., $8 profit is earned by a retailer as a profit of the sold item (a).

As mentioned in Table 3, we are using the utility values as input vector which are obtained by using the *Example*1. In this research we are using the horizontal federated learning method where data distribution is different by feature space is same as shown in Table 3, the transaction has fixed 5 items. Our goal is to map the transaction data by learning a function $f : \mathcal{D} \rightarrow \mathbb{R}^d$ such that every transaction $T_i \in \mathcal{D}$ is mapped to a d - dimensional continuous vector. The learning requires having the similarity among the transactions in a way that correlated items per transaction should have similar embedding. The embedding $\mathbf{X} = [f(T_1), f(T_2), \ldots, f(T_N)]$ is then used by the dynamic clustering method to select dynamic cluster as mentioned in Table 4 and illustration is mentioned in Fig. 1. As seen in Table 4, three distinct cluster are constructed (illustration for understanding), *cluster*1 represents the high utility ($90 - $97) transaction belonging to *customer*1, *cluster*2 have the mid utility value, whereas *cluster*3 represented in brown colour has minimum utility.

3.1. Client-Server-Based Federated Learning

In the Federated learning method, model customized weights are moved globally (through all the edge clients connected to the server). After receiving the optimized weights, the client trained the model locally (at the client end on the network's edge). The

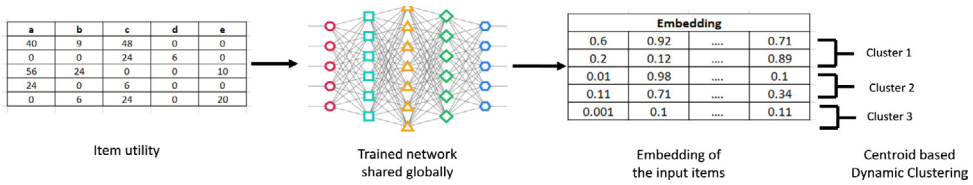
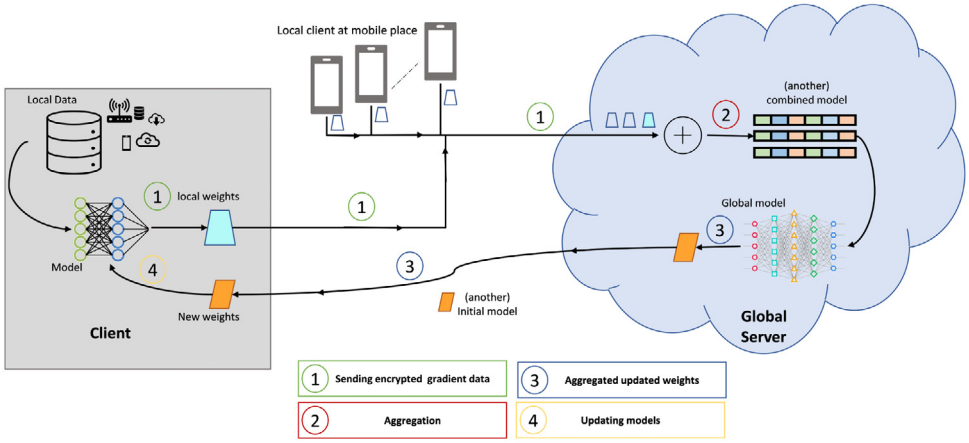
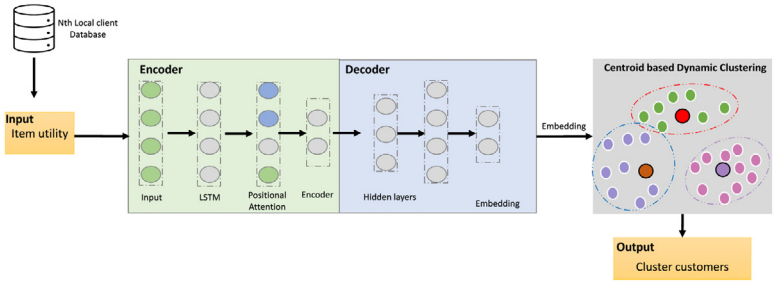


Fig. 1. The illustration of the proposed approach.



(a) An attention-based method was trained on the transaction data of multiple stores. The centralized model aggregate server for multiple clients without exchanging sensitive information. The attention network can adopt the global weights and gradient from the client's local train models. (clients).



(b) The local level processing of item utility value, used as a feature. The encoder layer converts the utility item into low dimensional representations. The proposed CBDC algorithm cluster the embedding into segments.

Fig. 2. The global framework of the designed attention-based federated learning for transaction embedding.

Table 4
Expected output (colour highlighted) of the clustering method C_{ID}: customer identity, T_{ID}: transaction identity.

C_{ID}	T_{ID}	Utility
C_1	T_1	97
C_2	T_2	38
C_1	T_3	90
C_2	T_4	32
C_3	T_5	58

mobile edge clients (connected with the server) can benefit from on-device data without sharing it across the network. We use this concept in this research; the server is an online computational efficient system where mobile edge clients may be local store devices (mobile point of sales devices)[2] in this way system able to train intelligent systems by keeping the privacy of any individual.

We mention the framework for the multiple stores in Fig. 2. Each client data exhibits non-overlapping distribution, local model (model at client end), and database. The data distribution is varied among the client randomly. Given these differences, we use UK store data where stores multiple branches are placed in different cities and countries. We connect the store database via a distributed network. The stores required customer analysis to give the royalty packages and analyse the product demands among different communities. The environment is non-independent, and distributed system settings are identical. The server–client environment is used by the proposed federated learning method. We mention that only an attention-based deep learning model is shared among the node; the clustering is performed locally by each client. Local store owners do customer analysis or community detection. However, the low dimensional embedding is created for the semantic similarity of product and customers. This helps the dynamic cluster method to form the community of high to low valued customers.

2 https://squareup.com/shop/hardware/ca/en.

In the proposed model, the client/store data are kept in the client/store database. The model is trained locally on each iteration, and then weights/gradient are transferred to the server. The server receives the model and then aggregates the weights of all the clients. This helps the model learn the low-dimensional features and transfer them to all the clients connected to the network. The purpose of the transfer is that it helps the model to improve the predictive ability and move towards the generalized model. The global model then uses the aggregate weights to update the global model. The server performs the aggregation, and then the updated model is shared with the client with global weights. After a certain iteration of the rounds, clients reach the convergence point. Then, the final global sharing is performed among the clients. During the experimental evaluation, we set the round to 20 and epochs per client 50. The client can select the global aggregated model or the best local iteration-based data. After getting the low dimensional feature embeddings, we do the dynamic cluster. For optimization and hyperparameter tuning, we set the initial stopping patient value to 10. In our empirical analysis, if we set the embedding size to a higher number, the model performance improved. It is because the decoder able to map positional attention to the bigger vector. The bigger vector space helps to reduce the loss of important information. The averaging of the global weights help to reduce the loss from the weighted combination of K losses $\{\mathcal{L}_k\}_{k=1}^{K}$ (client losses) of the distributed aggregated function [39]. The task of the model is to find the parameter ϕ that minimizes the L when given the local data X_k where X combination of local data sets and representation of the embedding. The coefficient $W_K > 0$ denotes the weights of the client K model. The model K is trained on local data, and only weights are distributed among the server. The weight is aggregated by summing the number of clients in the network. Eq. (3) represents the loss function [9].

$$\mathcal{L}(X;\phi) = \sum_{k=1}^{K} w_k \mathcal{L}_k\left(X_k;\phi\right) \tag{3}$$

3.2. Attention network

We used the transaction data and utility values as an input feature for each item/product. We mention the calculation in Table 3. The input feature is only the item utility value for the transaction. The utility is the product of quantity and profit value, as described in the previous section. We use these values as contextual information for the attention-based encoder–decoder model. After that, we use the attention mechanism to apply to the contextual features; this results in the embedding of a reducing set. The reduced embedding carries semantic similarity among the items. We used unique structures from deeper to wider networks, as mentioned in Fig. 3. The embedding [200, 100, 50, 100, 200] is optimized as it has low error rates and low dimension. The dense layer has several Relu units. We used the Luong attention method that uses the decoder hidden state [40]. The positional attention is calculated with the hidden state of the decoder. We then passed it to the dense network of different hidden layers, having the Adam optimizer for the learning rate. As mentioned in Table 3, we padded the data with zero vector if the item is not present. The network is trained on 20 rounds and 50 epochs per client (early's stopping set to 10). Each transaction item utility value, an input vector, and low dimension semantic embedding are created.

3.3. Centroid-based dynamic clustering (CBDC)

Given transaction data, the trained embedding model, we start by converting the transaction utility data into low dimensional space *(Algorithm 1 — line 1)*. Then, we partition low dimensional space randomly (initialization), each with the mean centroid. We reassign the cluster by computing the semantic similarity determined by centroid based on the low dimensional transaction embedding *(Algorithm 1 — lines 4 to 11)*. Then, the next iteration repeats until the centroid converges or not changed. The clusters are just collection; we can define the centroid of the cluster by computing the average of the transaction belonging to each cluster; thus, if $S_1, S_2, \ldots, S_N$ are transaction belonging to cluster and centroid *(Algorithm 1 — lines 16 to 19)* and illustrated in Fig. 4. All the transactions in the cluster except for the transaction with which the cluster centroid is being compared. Thus, we have transaction $S_1, S_2, \ldots, S_N$ and we want similarity between clusters and transaction appearing in the cluster, we determine the cluster centroid using the $S_1 \cup S_2 \cup \cdots \cup S_{G-1} \cup_{G+1} \cup \cdots \cup S_N$, we omit S_G in calculating the cluster centroid *(Algorithm 1 — line 15)*.SM_N is the embedding belonging to cluster K $N = 1 \ldots j$ where j is the number of transaction belonging to cluster K. The reduced function is applied if the cluster has less than five

Algorithm 1 Centroid-based Dynamic clustering (CBDC).

INPUT: Trained embedding E, transaction data $T = \{T_i \mid i = 1, \ldots, N\}$.
OUTPUT: *Set of Clusters*

```
1:  S_V ← E(T)   ▷ Convert transaction utility item value to low dimensional semantic embedding
2:  Partition the S_V, into K clusters randomly
3:  for all i ∈ N do
4:      for all k ∈ K do
5:          if i ≤ N then
6:              k ← k + 1
7:              SV_k ← SIM(T_i)
8:          else
9:              k ← 1
10:             SV_k ← SIM(T_i)
11:         end if
12:     end for
13:     repeat until convergence
14:     Find the mean centroid for each cluster
15:     for all k ∈ K do
16:         similarity ← SIM(M_k, SM_N)
17:         Re-assign each transaction to the cluster
18:         corresponding to the cluster centroid to which it is
19:         closest (semantically similar).
20:         Re-assign (S, M)
21:         Reduce (S, M)
22:     end for
23: end for
24: Return Set of Clusters
```

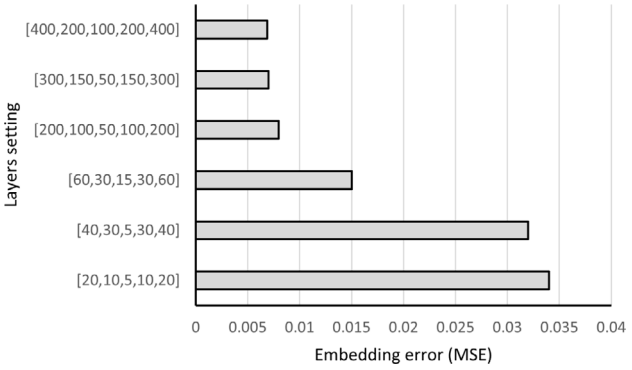


Fig. 3. Embedding structure and error rate.

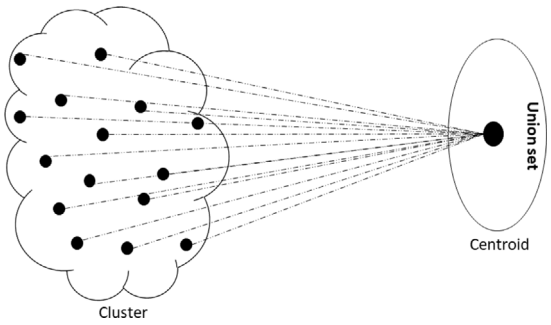


Fig. 4. Centroid method illustration.

instances and merged with closed centroid *(Algorithm 1 — line 22)*. The method illustrated in Fig. 4. For similarity among vectors, we used the cosine similarity mentioned as in Eq. (4). Given two vectors t, e, where t representing transaction embedding and t as centroid. We computer cosine similarity to compare centroid and transaction embedding two vectors.

$$(\mathbf{t}, \mathbf{e}) = \frac{\text{te}}{\|\mathbf{t}\|\|\mathbf{e}\|} = \frac{\sum_{i=1}^{n} \mathbf{t}_i \mathbf{e}_i}{\sqrt{\sum_{i=1}^{n} (\mathbf{t}_i)^2}\sqrt{\sum_{i=1}^{n} (\mathbf{e}_i)^2}}. \tag{4}$$

4. Experimental evaluation

For experiments, we use datasets from Kaggle.[3] A UK-based online store has been providing the sales data for different products for the period of one year (Nov 2016 to Dec 2017) as mentioned in Table 5. Four thousand three hundred seventy-two customers have 541909 transactions for the past two years. The online platform sells gifts. A small business or sub-organization is associated with the store that bought the product in bulk and sold it to other customers through the retail outlet channel. Different customers also bought the product directly from the store. The online store wants to cluster a group of customers who buy high purchases of certain products. The objective is to use the pattern analysis method to detect the customer product relationship and roll out the loyalty program. Customer segmented groups help to identify the useful patterns between the type of customer and products. In our research, we use the case study to demonstrate the effectiveness of our proposed model. Where federated learning *Centroid-based dynamic clustering (CBDC)* is being evaluated with and without an attention network. The federated learning approach mentioned in Tables 6 and 7, the method is evaluated using cluster size two and without an attention network. A recurrent neural network (RNN) with gated recurrent units (GRU) is used for embedding creation without attention method [41]. We used LSTM unidirectional architecture with the element-wise average method on the last layer [41]. LSTM cells with RNN architecture performed well. The LSTM cells help to preserve the information for long transaction utility values. CBDC used the federated learning method with an attention network. We evaluated the evaluated by using the cluster K size of 2, 4, 8, 16, 32, 64.

4.1. Performance metrics

The methods were evaluated using the Area Under the Receiver Operating Characteristic Curve (ROC AUC). The ROC was produced using the true positive rate (sensitivity) and false-positive rate (specificity) at thresholds ranging between 0–1. The confusion matrix helps to compute the rates. The ROC curve, TPR, and FPR are calculated based on the confusion matrix.

$$\text{TPR} = \frac{\text{\# of true transaction in the cluster}}{\text{\# of actual transaction in the cluster}}. \tag{5}$$

$$\text{FPR} = \frac{\text{\# of false transaction in the cluster}}{\text{\# of actual transaction in the cluster}}. \tag{6}$$

4.2. Results discussion

In this research, the goal is to detect the royal customer or set of customers. The clustering is the critical step in the framework. The same customer concerning buying behaviour was required to group. As a result, the weight sharing embedding method is adopted, and dynamic clustering is performed at individual stores. The federated learning method and local store model concerning different communities are compared. As we are using the horizontal federated learning, the feature values (utility item) remain the same; however, we used 50 stores random distribution and 15 stores fixed distribution to check the robustness of the model. For both distributions, we use the optimized model from Fig. 3 having a mean square error of 0.006 and embedding vector 200.

As mentioned in Table 6, we clustered the 4372 customers with 541909 transactions into various communities and communication rounds by randomly distributed the data to 50 stores. The federated learning model tends to have a lower score than the communities number set to 2 and 4. The reason is that a low number of clusters has fewer errors in distribution. The homogeneous cluster, i.e., each cluster only instances from a single class and complete cluster, i.e., all instances from the single class are to a single cluster are rarely achieved compared with higher cluster numbers. However, the model able to perform better in detecting relevant communities. The community with cluster 4 achieved 0.72 ROC under 25 rounds of communication.

Similarly, in Table 7, we clustered the 4372 customers with 541909 transactions into various communities and communication rounds by the fixed distribution of the data to 15 stores. We observed that the performance dropped from 2% with fixed stores and data distribution as the number of instances is mapped to the fixed stores. The communities detected by the methods have no distinct boundary and overlapped instances. As a result, Federated learning achieved 0.68 with no much drop in the performance, whereas communities with dynamic clustering methods have drop performance from 0.75 to 0.7, and convergence round also increased. Therefore, the data distribution method is a critical step for federated learning methods. For each communication round, all clients are required to do 50 epochs. The early stopping method is set to avoid overfitting issues. We added Tables 6 and 7 for the communication round comparison per 50 epochs. The model requires more communication rounds to improve. However, with each round, the improvement becomes very slow. As a result, we set the stops for further communications.

Fig. 5 shows that lower clustering numbers result in higher values, and a higher clustering number results in lower ROC value. This is due to homogeneous and complete cluster properties. When the number of clusters is large, the model cannot distinguish the clear boundaries. Lower cluster count leads to more distinct boundaries. The separation of the cluster is easily recognized when communities are 2, 4, 8 as they have a large instance than the rest. The random distribution is also a key factor. For realistic comparison, the model data are distributed randomly. However, the country-based distribution could result in better performance. The federated learning achieved 0.68 for the 50 rounds, whereas the other communities with cluster size 2 achieved 0.75. This reason depicts that inconsistent splitting of distribution leads to less effective learning. However, both models

[3] https://www.kaggle.com/vik2012kvs/high-value-customers-identification.

Table 5
UK based store data set (countries wise orders).

Country	Orders	Country	Orders	Country	Orders	Country	Orders
United Kingdom	356727	Norway	1086	USA	291	RSA	58
Germany	9480	Italy	803	Israel	247	Lebanon	45
France	8475	Islands	757	Unspecified	241	Lithuania	35
EIRE	7475	Finland	695	Singapore	229	Brazil	32
Spain	2528	Cyprus	611	Iceland	182	Czech Republic	30
Netherlands	2371	Sweden	461	Canada	151	Bahrain	17
Belgium	2069	Austria	401	Greece	146	Saudi Arabia	10
Switzerland	1877	Denmark	389	Malta	127		
Portugal	1471	Japan	358	United Arab Emirates	68		
Australia	1258	Poland	341	European Community	61		

Table 6
The convergence of the customer clustering with ROC performance measure, training data distribution from randomly chosen 50 stores (clients). *(-) indicated method converges after rounds, K indicates number of clusters (number of communities), and FL is Federated learning approach without attention method.*

Method	Communication round					
	5	10	15	20	25	50
FL: k = 2	0.45	0.47	0.55	0.65	0.68	0.68
CBDC: k = 2	0.41	0.45	0.65	0.7	–	–
CBDC: k = 4	0.4	0.43	0.64	0.69	0.72	–
CBDC: k = 8	0.39	0.48	0.61	0.64	0.68	–
CBDC: k = 16	0.41	0.43	0.56	0.6	0.61	–
CBDC: k = 32	0.43	0.43	0.58	–	–	–
CBDC: k = 64	0.41	0.55	0.58	–	–	–

Table 7
Convergence table to communication rounds in the customer clustering, communities were trained, tested and compared with 15 stores fixed store distributed data. *(-) indicated method converges after rounds, K indicates number of clusters (number of communities), and FL is Federated learning approach without attention method.*

Method	Communication round					
	5	10	15	20	25	50
FL: k = 2	0.39	0.45	0.53	0.63	0.66	0.68
CBDC: k = 2	0.39	0.43	0.63	0.68	–	–
CBDC: k = 4	0.38	0.41	0.62	0.67	0.7	–
CBDC: k = 8	0.37	0.46	0.59	0.62	0.66	–
CBDC: k = 16	0.39	0.41	0.54	0.58	0.59	–
CBDC: k = 32	0.39	0.41	0.56	–	–	–
CBDC: k = 64	0.39	0.53	0.56	–	–	–

converged significantly faster, with fewer communication rounds. The early stopping method was adopted to stop training after the repetition of error for the number of times to reduce overfitting. As a result, some model does not take part in the communication after they achieved convergence.

In Fig. 6, an F-measure comparison is performed that represents the clustering comparison between 15 and 50 stores randomly distributed data. When the communication rounds and the number of clusters are increased, the model tends to have a lower performance, whereas when fewer clusters are made, the model performs better. This includes that there are few clusters in the market analysis. The company should set the clustering size between 4 to 8 to find optimal and accurate results to find royal customers.

4.3. Dynamic clustering analysis

We control the clustering method by simple configurations. There are a few parameters that are required to be tuned. As a result, we can adapt the method in different fields. The embeddings can be replaced with different architectures. Also, the size and shape of the feature can be adjusted according to domain requirements. We can enrich the features with support, frequency, and occupancy of the item or products. The first stopping mechanism helps to reduce over-fitting results robust model for the network. The attention mechanism and context vector helps to create distinct semantic aware embedding that results in the improved cluster boundary and centroid calculation. The clustering method of embedding is excellent for distributed learning. Many applications required clustering and supervised learning tasks where data labelling can do through the dynamic cluster method.

4.4. Future work

In the future, in addition to the federated learning method, we will apply active learning. The additional features for item value, i.e., uncertainty, utility, frequency, and co-occurrence, can be used as features. We also plan to do a distributed data analysis. Our empirical analysis suggests that client data scalability analysis should be done during the federated learning method. The information must be shared regarding instance correlation (linear and non-linear relation), dimension size, and number instances. The modality analysis should also be performed to conduct to get the uni-modal and modal distribution. The noise and contamination (anomalies) analysis required to be considered as it contaminated the data and result in the under performance of models connected through networks.

5. Conclusion

Real-time streaming data has definitive privacy concerns, especially in retail sales data where transactions have high intrinsic value and are loaded with personal information. We have presented a novel federated learning method with a dynamic clustering method to cluster customers based on purchase behaviour that can be administered at the edge of mobile sales networks using edge Point of Sales devices. Our developed model has achieved high performance. The method is helpful for the data labelling task as well as clustering analysis. Our model was evaluated against federated learning by using ROC as a performance measure. ROC measures the true positive rate and false positive of data instance. From experimental results, it is seen that attention-based federated learning has an advantage over the traditional approach. The model can do better locally while communicated fewer rounds, and the method converges quickly.

CRediT authorship contribution statement

Usman Ahmed: Theoretical development, Drafting the whole article. **Gautam Srivastava:** Experimental design and analysis, Revising and proofread article. **Jerry Chun-Wei Lin:** Conceptualization and theoretical development, Reviewing the contents of the article, Approving the final version.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

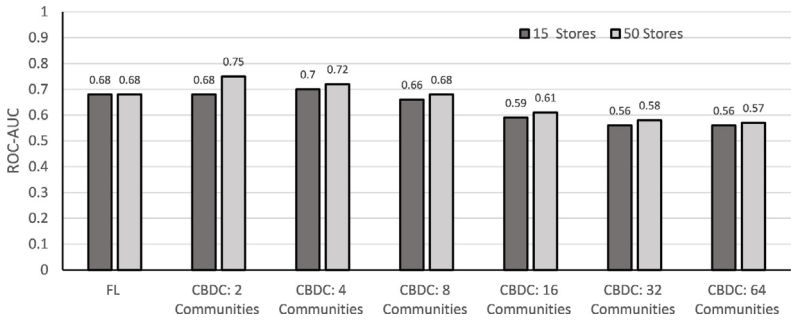


Fig. 5. ROC-AUC values trained, tested and compared with fixed 15 stores and 50 stores (clients) randomly distributed data.

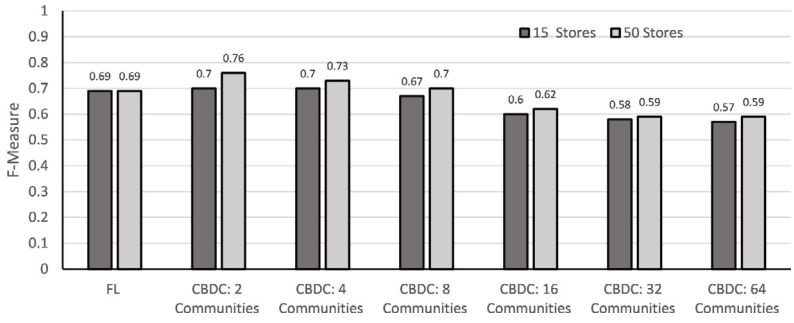


Fig. 6. F-measure compared with fixed 15 stores and 50 stores (clients) randomly distributed data.

References

[1] W. Gan, J.C.W. Lin, P. Fournier-Viger, H.C. Chao, P.S. Yu, HUOPM: High-utility occupancy pattern mining, IEEE Trans. Cybern. 50 (3) (2020) 1195–1208.

[2] W. Gan, J.C.W. Lin, P. Fournier-Viger, H.C. Chao, V. Tseng, P. Yu, A survey of utility-oriented pattern mining, IEEE Trans. Knowl. Data Eng. 33 (4) (2019) 1306–1327.

[3] M. Hu, Cambridge analytica's black box, Big Data Soc. 7 (2) (2020) 205.

[4] D. Połap, G. Srivastava, K. Yu, Agent architecture of an intelligent medical system based on federated learning and blockchain technology, J. Inf. Secur. Appl. 58 (2021) 102748.

[5] V. Mothukuri, R.M. Parizi, S. Pouriyeh, Y. Huang, A. Dehghantanha, G. Srivastava, A survey on security and privacy of federated learning, Future Gener. Comput. Syst. 115 (2021) 619–640.

[6] U. Ahmed, G. Srivastava, J.C.W. Lin, A machine learning model for data sanitization, Comput. Netw. 189 (2021) 107–914.

[7] U. Ahmed, J.C.W. Lin, G. Srivastava, Y. Djenouri, A deep Q-learning sanitization approach for privacy preserving data mining, in: Adjunct Proceedings of the 2021 International Conference on Distributed Computing and Networking, 2020, pp. 43–48.

[8] K. Chang, N. Balachandar, C. Lam, D. Yi, J. Brown, A. Beers, B. Rosen, D.L. Rubin, J. Kalpathy-Cramer, Distributed deep learning networks among institutions for medical imaging, J. Am. Med. Inform. Assoc. 25 (8) (2018) 945–954.

[9] H.R. Roth, K. Chang, P. Singh, N. Neumark, W. Li, V. Gupta, S. Gupta, L. Qu, A. Ihsani, B.C. Bizzo, et al., Federated learning for breast density classification: A real-world implementation, in: Domain Adaptation and Representation Transfer, and Distributed and Collaborative Learning, Springer, 2020, pp. 181–191.

[10] W. Gan, J.C.W. Lin, P. Fournier-Viger, H.C. Chao, T.-P. Hong, H. Fujita, A survey of incremental high-utility itemset mining, WIREs Data Min. Knowl. Discov. 8 (2) (2018).

[11] H. Mohammed, N. Clarke, F. Li, Evidence identification in heterogeneous data using clustering, in: The International Conference on Availability, Reliability and Security, 2018, pp. 35.

[12] Y. Chen, Z. Wei, X. Huang, Incorporating corporation relationship via graph convolutional neural networks for stock price prediction, in: ACM International Conference on Information and Knowledge Management, 2018, pp. 1655–1658.

[13] M. Kalra, N. Lal, S. Qamar, K-mean clustering algorithm approach for data mining of heterogeneous data, in: Information and Communication Technology for Sustainable Development, 2018, pp. 61–70.

[14] K.P. Sinaga, M.-S. Yang, Unsupervised K-means clustering algorithm, IEEE Access 8 (2020) 80716–80727.

[15] J. Cai, H. Wei, H. Yang, X. Zhao, A novel clustering algorithm based on DPC and PSO, IEEE Access 8 (2020) 88200–88214.

[16] J. Lin, Y. Shao, Y. Djenouri, U. Yun, ASRNN: A recurrent neural network with an attention model for sequence labeling, Knowl.-Based Syst. 212 (2021) 106548.

[17] A. Belhadi, Y. Djenouri, D. Djenouri, T. Michalak, J.C.W. Lin, Deep learning versus traditional solutions for group trajectory uutliers, IEEE Trans. Cybern. (2020) 1–12.

[18] G. Nguyen, S. Dlugolinsky, M. Bobák, V.D. Tran, A.L. García, I. Heredia, P. Malík, L. Hluchý, Machine learning and deep learning frameworks and libraries for large-scale data mining: A survey, Artif. Intell. Rev. 52 (1) (2019) 77–124.

[19] K. Cho, B. van Merrienboer, Ç. Gülçehre, D. Bahdanau, F. Bougares, H. Schwenk, Y. Bengio, Learning phrase representations using RNN encoder-58decoder for statistical machine translation, in: A. Moschitti, B. Pang, W. Daelemans (Eds.), The Conference on Empirical Methods in Natural60Language Processing, 2014, pp. 1724–1734.

[20] M. Siam, S. Elkerdawy, M. Jägersand, S.K. Yogamani, Deep semantic segmentation for automated driving: Taxonomy, roadmap and challenges, in: IEEE International Conference on Intelligent Transportation Systems, 2017, pp. 1–8.

[21] V. Sze, Y.H. Chen, T.J. Yang, J.S. Emer, Efficient processing of deep neural networks: A tutorial and survey, Proc. IEEE 105 (12) (2017) 2295–2329.

[22] M. Wainberg, D. Merico, A. Delong, B.J. Frey, Deep learning in biomedicine, Nature Biotechnol. 36 (9) (2018) 829–838.

[23] G.V. Horn, O.M. Aodha, Y. Song, Y. Cui, C. Sun, A. Shepard, H. Adam, P. Perona, S.J. Belongie, The Inaturalist species classification and detection dataset, in: IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 8769–8778.

[24] P. Rajpurkar, J. Irvin, K. Zhu, B. Yang, H. Mehta, T. Duan, D.Y. Ding, A. Bagul, C. Langlotz, K.S. Shpanskaya, M.P. Lungren, A.Y. Ng, Chexnet: Radiologist-level pneumonia detection on chest X-Rays with deep learning, CoRR (2017) arXiv:1711.05225.

[25] Y. Wu, M. Schuster, Z. Chen, Q.V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, J. Klingner, A. Shah, M. Johnson, X. Liu, L. Kaiser, S. Gouws, Y. Kato, T. Kudo, H. Kazawa, K. Stevens, G. Kurian, N. Patil, W. Wang, C. Young, J. Smith, J. Riesa, A. Rudnick, O. Vinyals, G. Corrado, M. Hughes, J. Dean, Google's neural machine translation system: Bridging the gap between human and machine translation, CoRR (2016) arXiv:1609.08144.

[26] N.P. Jouppi, et al., In-datacenter performance analysis of a tensor processing unit, in: Proceedings of the 44th Annual International Symposium on Computer Architecture, 2017, pp. 1–12.

[27] H.I. Fawaz, Deep learning for time series classification, CoRR (2020) arXiv:2010.00567.

[28] J. Lu, J. Yang, D. Batra, D. Parikh, Hierarchical question-image co-attention for visual question answering, in: D.D. Lee, M. Sugiyama, U. von Luxburg, I. Guyon, R. Garnett (Eds.), Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems, 2016, pp. 289–297.
[29] D. Bahdanau, K. Cho, Y. Bengio, Neural machine translation by jointly learning to align and translate, in: Y. Bengio, Y. LeCun (Eds.), The International Conference on Learning Representations, 2015, arXiv:1409.0473.
[30] K. Xu, J. Ba, R. Kiros, K. Cho, A.C. Courville, R. Salakhutdinov, R.S. Zemel, Y. Bengio, Show, attend and tell: neural image caption generation with visual attention, in: F.R. Bach, D.M. Blei (Eds.), The International Conference on Machine Learning, in: JMLR Workshop and Conference Proceedings, vol. 37, 2015, pp. 2048–2057.
[31] T. Luong, H. Pham, C.D. Manning, Effective approaches to attention-based neural machine translation, in: L. Màrquez, C. Callison-Burch, J. Su, D. Pighin, Y. Marton (Eds.), The Conference on Empirical Methods in Natural Language Processing, 2015, pp. 1412–1421.
[32] Q. Yang, Y. Liu, T. Chen, Y. Tong, Federated machine learning: concept and applications, ACM Trans. Intell. Syst. Technol. (TIST) 10 (2) (2019) 12:1–12:19.
[33] R. Shokri, V. Shmatikov, Privacy-preserving deep learning, in: ACM SIGSAC Conference on Computer and Communications Security, 2015, pp. 1310–1321.
[34] J. Hayes, O. Ohrimenko, Contamination attacks and mitigation in multi-party machine learning, CoRR (2019) arXiv:1901.02402.
[35] M. Fredrikson, S. Jha, T. Ristenpart, Model inversion attacks that exploit confidence information and basic countermeasures, in: ACM SIGSAC Conference on Computer and Communications Security, 2015, pp. 1322–1333.
[36] P. Mohassel, P. Rindal, ABY^3: A mixed protocol framework for machine learning, in: D. Lie, M. Mannan, M. Backes, X. Wang (Eds.), ACM SIGSAC Conference on Computer and Communications Security, 2018, pp. 35–52.
[37] I. Hegedűs, G. Danner, M. Jelasity, Decentralized learning works: An empirical comparison of gossip learning and federated learning, J. Parallel Distrib. Comput. 148 (2021) 109–124.
[38] M. Chamikara, P. Bertok, I. Khalil, D. Liu, S. Camtepe, Privacy preserving distributed machine learning with federated learning, Comput. Commun. 171 (2021) 112–125.
[39] B. McMahan, E. Moore, D. Ramage, S. Hampson, B.A. y Arcas, Communication-efficient learning of deep networks from decentralized data, in: Artificial Intelligence and Statistics, 2017, pp. 1273–1282.
[40] M.T. Luong, H. Pham, C.D. Manning, Effective approaches to attention-based neural machine translation, 2015, arXiv preprint arXiv:1508.04025.
[41] U. Ahmed, S.K. Mukhiya, G. Srivastava, Y. Lamo, J.C.-W. Lin, Attention-based deep entropy active learning using lexical algorithm for mental health treatment, Front. Psychol. 12 (2021).

Usman Ahmed is a Ph.D. candidate at the Western Norway University of Applied Sciences (HVL). He has rich experience in building and scaling high-performance systems based on data mining, natural language processing, and machine learning. Usman's research interests are sequential data mining, heterogeneous computing, natural language processing, recommendation systems, and machine learning. He has completed the Master of Science degree in computer science at Capital University of Science and Technology, Islamabad, Pakistan. Usman Ahmed was awarded a gold medal for his bachelor of computer science degree from Heavy Industries Taxila Education City.

Gautam Srivastava was awarded his B.Sc. degree from Briar Cliff University in the U.S.A. in the year 2004, followed by his M.Sc. and Ph.D. degrees from the University of Victoria in Victoria, British Columbia, Canada in the years 2006 and 2012, respectively. Dr. G, as he is popularly known, is active in research in the field of Cryptography, Data Mining, Security and Privacy, and Blockchain Technology. In his 5 years as a research academic, he has published a total of 45 papers in high-impact conferences in many countries and in high-status journals (SCI, SCIE). He is an IEEE Senior Member.

Jerry Chun-Wei Lin received the Ph.D. degree from the Department of Computer Science and Information Engineering, National Cheng Kung University, Tainan, Taiwan. He is the full Professor with the Department of Computer Science, Electrical Engineering and Mathematical Sciences, Western Norway University of Applied Sciences, Bergen, Norway. He has published more than 400+ research articles in refereed journals (IEEE TKDE, IEEE TFS, IEEE TNNLS, IEEE TCYB, IEEE TII, IEEE TITS, IEEE TNSE, IEEE TETCI, IEEE SysJ, IEEE SensJ, IEEE IOTJ, ACM TKDD, ACM TDS, ACM TMIS, ACM TOIT, ACM TIST, ACM TOSN) and international conferences (IEEE ICDE, IEEE ICDM, PAKDD, PKDD). His research interests include data mining, soft computing, artificial intelligence and machine learning, and privacy-preserving and security technologies. He is also the project co-leader of well-known SPMF: An Open-Source Data Mining Library, which is a toolkit offering multiple types of data mining algorithms. He also serves as the Editor-in-Chief of the International Journal of Data Science and Pattern Recognition. He is the Fellow of IET, Senior Member for both IEEE and ACM.