Engineering Applications of Artificial Intelligence 127 (2024) 107215

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

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

Towards an autonomous clinical decision support system

Sapir Gershov [a,*], Aeyal Raz [b], Erez Karpas [c], Shlomi Laufer [c]

[a] Technion Autonomous Systems Program, Haifa, Israel
[b] Rambam Health Care Campus, Haifa, Israel
[c] The Faculty of Data and Decisions Science, Technion, Haifa, Israel

ARTICLE INFO

Keywords:
Clinical decision support systems
Medical knowledge
Decision-making
Simulation
Graph networks

ABSTRACT

Clinicians' decision-making is of utmost importance during critical situations. Thus, integrating Clinical Decision Support Systems (CDSS) may assist the medical staff by enhancing the decision-making process, eventually improving patient outcomes. The potential of an autonomous CDSS, proficient in predicting and guiding medical treatment, is significant—especially in situations where every second counts.

We proposed a methodology to design a CDSS based on observational data of clinical procedures. This approach employs graph-convolutional networks (GCN) to encapsulate medical knowledge from simulated clinical procedures with sequential data. Consequently, our model can extrapolate from these procedures, identifying novel structural and characteristic combinations. This innovative method harnesses information that might elude human observers. Moreover, our model generates action sequences that a human physician has not previously executed.

Traditional techniques tend to fall short in adapting to changing trends, thus failing to anticipate human actions. Conversely, advanced models like GCN have demonstrated promising potential in tasks like human action prediction, including activity recognition. We assessed these performances using benchmark datasets, which yielded encouraging results.

Additionally, we constructed a graph-based CDSS to deliver pertinent medical advice. We outline a methodology to monitor the procedure's current stage and predict the physician's subsequent action, facilitating time-saving measures like pre-emptive instrument preparation. Our novel CDSS methodology achieved an F_1-score of 0.899 and 0.714 when performing one and two-step predictions, respectively. Furthermore, our simulations illustrate a considerable time-saving potential, with an average reduction of approximately 00:01:28 ± 00:01:15 min in the preparation time for adrenaline dosage, a crucial component for successful resuscitation.

1. Background

1.1. Graph convolutional network

In the last decade, Convolutional Neural Networks (CNNs) have gained extensive achievements in Euclidean data. However, data in the real world may have underlying graph structures that are non-Euclidean. The non-regularity of data structures has led to recent advancements in new forms of CNNs (Wu et al., 2019).

Graph Convolutional Networks (GCNs) (Kipf and Welling, 2016) are an evolved form of CNNs on graphs, which have already achieved state-of-the-art results in various application areas (Zhang et al., 2019). Instead of having an input of 2-D or 3-D arrays, GCN takes a graph as an input. For these models, the goal is to learn a function of signals/features on a graph $G = (V, E)$, which takes as input:

- A feature description x_i for every node i; summarized in a $N \times D$ feature matrix X, where N is the number of nodes and D is the number of input features.
- A representative description of the graph structure in matrix form; typically in the form of an adjacency matrix A.

This produces a node-level output Z (an $N \times F$ feature matrix, where F is the number of output features per node). Similar to CNNs, a k-layer GCN is identical to applying a k-layer convolution on the feature vector x_i of each node in the graph. Every layer can then be written as a non-linear function $H^{(l+1)} = f(H^{(l)}, A)$, with $H^{(0)} = X$ and $H^{(L)} = Z$, L being the number of layers. The specific models differ in how $f(\cdot, \cdot)$ is chosen and parameterized.

The "graph convolution" applies the same linear transformation to all node's neighbors. The difference is in the hidden representation of

* Corresponding author.
E-mail address: sapirgershov@campus.technion.ac.il (S. Gershov).

https://doi.org/10.1016/j.engappai.2023.107215
Received 7 May 2023; Received in revised form 4 September 2023; Accepted 26 September 2023
Available online 4 October 2023

each node, as it is normalized with its neighbors at the beginning of each layer. Afterward, the network can learn the graph representations by stacking layers of filters followed by a nonlinear activation function. This is a low-dimensional representation of the entities and relations in the graph. They provide a generalizable context about the overall graph that can be used to infer relations.

The input to graph convolution layer is a set of N node features from embedding layer $\mathbf{h} = \{h_1, h_2, \ldots, h_N\}$ where $h_i \in \mathbb{R}$ represents the d-dimensional features of ith node; a set of relation types $R = \{r_1, r_2, \ldots, r_k\}$; and a set of relation features $\mathbf{m} = \{m_1, m_2, \ldots, m_k\}$, where $m_k \in \mathbb{R}$ is the feature vector of rth-relation type of dimension d.

1.2. Graph embedding

Generally, graph embedding aims to encode nodes into low-dimensional space, such that similarity in the latent embedded space approximates similarity in the original high-dimensional graph while maintaining the graph structure. This can be achieved by solving an optimization problem with an unsupervised learning schema.

In line with the aforementioned definitions and notations, given a graph $G = (V, E)$, the task of learning graph node embeddings (e.g., L dimension, $L \ll |V|$) can be formulated as learning a projection ϕ, such that all graph nodes $(V = v_i | i = 1, 2, \ldots, |V|)$ can be encoded as embeddings from a high-dimensional space into a low-dimensional space. For this case, the node embedding form is deterministic point vectors: $\Phi = z_i \in \mathfrak{R}^L | i = 1, 2, \ldots, |V|$.

2. Introduction

In medical emergencies, practitioners grapple with diagnostic uncertainties and numerous interruptions within chaotic environments. Swift, coordinated strategies are vital for delivering optimal medical care amidst such challenges (Gaba et al., 2001; Banning, 2008). Compounding this is the complexity of medical data and patient information, which often present as vague, conflicting, non-interpretable, or even absent, making the clinical decision-making process more challenging (Begoli et al., 2019).

The idea of harnessing machine intelligence to assist in complex medical decision-making has gained traction among clinicians and Artificial Intelligence researchers alike in recent years (Yang et al., 2019). The intersection of these fields has given rise to advanced technologies, such as Clinical Decision Support Systems (CDSS), capable of providing situation-specific advice to medical staff, thereby optimizing patient care outcomes (Patel et al., 2009; Begoli et al., 2019). The evolution of machine learning has enabled computers to learn from past experiences and recognize patterns within clinical data. Consequently, future CDSS frameworks are anticipated to extract and interpret information that might have been otherwise overlooked or misinterpreted by humans (Sutton et al., 2020; Kawamoto et al., 2005). Moreover, they are expected to become increasingly autonomous, extending their functionality beyond suggestions to executing specific tasks independently (Challen et al., 2019).

Existing CDSS frameworks incorporate either reasoning with medical knowledge or sequential decision-making, both of which are strategies for reasoning under uncertainty. Most current state-of-the-art CDSS frameworks are built on knowledge graphs (KG) that amalgamate expert medical knowledge and clinical treatment guidelines. Schlichtkrull et al.'s 'R-GCN' (Schlichtkrull et al., 2018) paved the way for integrating graph networks for KG embedding. Consequently, numerous researchers have constructed heterogeneous graphs representing expert medical knowledge, also known as 'Medical KG'.

Medical KG are repositories of medicine-related information and clinical practice guidelines, constructed from large volumes of medical databases, demonstrating the potential to assist physicians in complex clinical decision-making (Ernst et al., 2015; Gong et al., 2021). However, current frameworks, such as MedGraph (Hettige et al., 2020) and SMR (Gong et al., 2021), have limitations, particularly in time-critical clinical situations.

Other works (Nemati et al., 2016; Prasad et al., 2017) have utilized sequential decision-making in clinical decision processes adaptable to neural networks, primarily through reinforcement learning (RL). However, these approaches are not strictly guided by medical knowledge and guidelines, posing potential safety and accountability concerns (Giordano et al., 2021). Nevertheless, several other works have demonstrated promising results by utilizing machine learning algorithms to diagnose patient clinical conditions (Fitriyani et al., 2020; Abdel-Basset et al., 2020; Kwan et al., 2020; Kim et al., 2021). These projects focused on creating tailor-made CDSS implementation that can take the patient's vital signs and the administered medications as input and advise the physicians on the recommended course of treatment. Their results indicated that the positive effects of the CDSS derive from compliance with clinical guidelines and integration with other clinical systems. Yet, it is essential to consider that CDSS are educational tools that may cause users to rely upon the CDSS for a specific task. Furthermore, this impact disrupts workflow and increases task completion time (Sutton et al., 2020).

In contrast, our work posits that a genuinely effective CDSS should incorporate embedded medical knowledge and sequential decision-making, particularly in medical emergencies. To date, the combined strengths of these strategies remain underexplored despite their individual areas being extensively researched. Systems integrating these crucial elements will be able to handle a wide range of decisions, especially in uncertain scenarios, and adequately assess the implications of proposed solutions in complex situations with numerous variables (Challen et al., 2019).

For this work, we propose to utilize Graph Convolution Network (GCN) models to incorporate these two elements. GCNs have a tremendous expressive power to learn the sequences representations and have achieved superior performance in embedding domain knowledge. To demonstrate the GCN's ability to understand and predict real-world human actions, we first tackle the Temporal Action Segmentation (TAS) task using two benchmark datasets of non-medical procedural activities.

Once we have demonstrated our GCN model robustness, we introduced a methodology to integrate embedded medical knowledge with sequential decision-making to create a fully autonomous CDSS suitable for real-time medical assistance. Our CDSS framework can interpret and reason with clinicians' workflow, providing autonomous support in standard and emergency procedures. This unique combination of knowledge and decision-making offers an improved CDSS framework compared to existing literature. We evaluated our approach on a real-life medical activity dataset we collected (Gershov et al., 2021, 2023). Our CDSS framework successfully predicted the need for adrenaline dosage and defibrillator shock preparation over a minute before the physician's request, underlining its practical utility and efficiency in a real-life setting.

In summary, this work proposes a framework for integrating CDSS to assist medical personnel by enhancing their decision-making process, eventually improving patient outcomes. Our work introduces several methodological innovations and contributions. Firstly, we proposed a technique for constructing a CDSS from observations of clinical procedures using graph convolutional neural networks to generalize from several simulated procedures. This methodology is suitable for fast-paced clinical execution, as seen in our reported results. Second, our proposed methodology incorporates two fundamental elements: embedded medical knowledge and sequential decision-making. Both are strategies for reasoning under uncertainty, thus improving our framework's resilience to poor data quality and ambiguous information. Third, our CDSS has been designed to harness the practitioners' verbal communication and "think-aloud" process. Thus, we created a system that does not disturb the clinicians' workflow. Lastly, we demonstrate a framework for tracking the current state of a medical procedure and predicting the physician's following actions.

Table 1
Example of the speech-based action recognition system output (translated from the original language).

Task	Timestamp	Evidence in text
Listening to the lungs and heart	00:03:00.520	"I am listening"
Adrenaline 0.2–0.3 mg	00:04:28.040	"Give me Adrenaline. OK, thank you. Adrenaline is inside."
Ventolin Inhalation - Half a cc of Ventolin	00:03:54.540	"We can do Ventolin inhalation"

3. Materials

We evaluated our proposed model on three datasets to demonstrate the GCN's potential to understand and predict real-world human actions. The first two are benchmark datasets for Temporal Action Segmentation (non-clinical), and the third contains actions executed by anesthesia residents (clinical).

3.1. Temporal action segmentation datasets

Temporal action segmentation (TAS) is a computer vision task aiming to segment a video where multiple actions occur sequentially. Each segment is a pre-defined action label (Zhao et al., 2017). State-of-the-art TAS models can distinguish hundreds of classes by exploiting two sources of information - video and text (Li et al., 2021; Singhania et al., 2022). The need to develop algorithms applicable to real-world scenarios has prompted the development of standardized datasets. These datasets are composed of videos where the participants execute a sequence of actions and are annotated with the start and end boundaries of action segments and their labels (Herath et al., 2017). In this work, we assessed our methodology for predicting the following procedural actions based on a given action sequence. Based on the available datasets, just two focus on goal-oriented, multi-step activities: '50Salads' (Stein and McKenna, 2013) and 'Assembly101' (Sener et al., 2022). We used the dataset's original action class sequences (i.e., textual), not their visual footage, as the input to our model.

50Salads: This dataset contains 50 videos of preparing two mixed salads. There are 17 fine-grained action classes.

Assembly101: The dataset contains 362 unique video sequences of people assembling and disassembling 101 toy vehicles. Assembly101 is annotated with over 1M action segments, bringing about 202 action classes. On average, each video features an average of 24 action classes.

3.2. Medical simulation dataset

In previous research (Gershov et al., 2021, 2023), we observed that the conversation among medical personnel in most cases might indicate the physical action being performed. Analyzing the participants' speech, they could automatically identify and fill in the appropriate rubrics in a task-specific checklist. To this end, we developed an end-to-end, fully automatic speech-based objective checklist validation system that identifies anesthesia residents' actions based solely on the participants' speech.

This system is designed to identify the action currently being executed by medical personnel and document it in the simulation checklist. In each filled-out checklist, the observed task (recognized action) comes alongside a timestamp representing the estimated time it was executed (Table 1).

We rely on the dataset collected by our system. The output of the speech-based action recognition system produces a series of actions which we refer to as the 'observed sequence of procedures'. Each action in the sequence is a task from the checklist, and the length of the 'observed sequence of actions' varies from 25–35 actions (repetition is possible). The checklist depicts the ideal 'observed sequence of actions' in two levels: (1) the correct actions required to treat the symptoms and (2) the optimal sequence of actions. Fig. 1 illustrates a real example of an 'observed sequence of procedures' for the VF scenario. As we can see, a few actions from the checklist have never been executed by the anesthesia residents in our dataset. In addition, there are several repetitions in the residents' sequence of actions, indicated by the edges' width and value. A more detailed description of the dataset and system is provided in our previous paper (Gershov et al., 2023).

As part of the data collecting process, we deployed four clinical scenarios: (1) patient with a severe anaphylaxis reaction; (2) postoperative patient with severe bradycardia; (3) postoperative patient with opioid overdose; (4) postoperative patient with severe hypoglycemia. For each scenario, a suitable detailed task-specific checklist was constructed. Each checklist included approximately 35 tasks, and each task execution quality was scored compared to standard medical guidelines. The checklist task score is scaled as follows: 0 for not observed, 1 for needs improvement, and 2 for meets expectations. When the simulation ended, the evaluator rated the overall resident's performance on a range of 1 – 5, where one is considered poor performance.

Fifty-two senior anesthesiology residents, 40 males and 12 females, participated in their study. Every participant performed both simulation scenarios (64 simulations in total).

We rely on the dataset collected by this system, the filled-out checklists, as a set of observed procedures, for the training and testing of the proposed model.

4. Methods

We now present the complete system diagram (see Figs. 2 and 3), and the following sections will explain the individual components.

4.1. Graph convolutional network - based link prediction

The link prediction problem is defined as follows. Given the node features X, the model can output whether an edge connects two nodes. To be more accurate, in a domain-specific graph $G(V, E)$ where $V = \{1, 2, \ldots, N\}$ is the node set and $E \subseteq V \times V$ is the link set, GCN utilizes edges $E \in G$ to aggregate and learn node embeddings. The possibility of a connection is decided according to the similarity score of two node embeddings (Kipf and Welling, 2016). Link prediction uses the resulting vectors to find possible and unobserved associations (links) between two nodes.

When applying the link prediction problem to a GCN model, the model optimizes the likelihood of connectivity between two nodes u and v, as a function of the node representation, ${h_u}^{(L)}$ and ${h_v}^{(L)}$, computed from the multi-layer GCN:

$$y_{u,v} = \phi({h_u}^{(L)}, {h_v}^{(L)}) \tag{1}$$

Where $y_{u,v}$ is the score between node u and node v. Given an edge connecting u and v, we encourage the $y_{u,v}$ score to be higher than the score between node u and a different node v' from graph G.

For this work, the link prediction is achieved via high-order heuristics to extract local enclosing subgraphs around links as the training data, thus allowing us to utilize a fully-connected neural network to learn which enclosing subgraphs correspond to link existence.

We will use Zhang and Chen's work (Zhang and Chen, 2018) to prove the following:

Definition 1 (*Enclosing Subgraph*)**.** For a graph $G = (V, E)$, given two nodes $x, y \in V$, the h-hop enclosing subgraph for (x, y) is the subgraph $G^h_{x,y}$ induced from G by the set of nodes $\{i | d(i, x) \leq h \vee d(i, y) \leq h\}$.

Since $G^h_{x,y}$ contains all h-hop neighbors of x and y, we than have the following theorem.

Theorem 1. *Any h-order heuristic for (x, y) can be accurately calculated from $G^h_{x,y}$.*

However, a very large h is needed for learning high-order heuristics. Thus, the following analysis proves that learning high-order heuristics is also feasible with a small h.

We support this first by defining the γ-decaying heuristic that, under certain conditions, can be well approximated from the h-hop enclosing subgraph. Furthermore, Zhang and Chen's work show that the most well-known high-order heuristics can be unified into this γ-decaying heuristic framework.

Definition 2 (*γ-decaying Heuristic*)**.** A γ-decaying heuristic for (x, y) has the following form:

$$H(x, y) = \eta \sum_{l=1}^{\infty} \gamma^l f(x, y, l) \tag{2}$$

where γ is a decaying factor between 0 and 1, η is a positive constant or a positive function of γ that is upper bounded by a constant, f is a non-negative function of x, y, l under the given network.

Theorem 2. *Give γ-decaying heuristic $H(x, y) = \eta \sum_{l=1}^{\infty} \gamma^l f(x, y, l)$, if $f(x, y, l)$ satisfies the following properties:*

- *$f(x, y, l) \leq \lambda^l$ where $\lambda < \gamma^{-1}$*
- *$f(x, y, l)$ is calculable from $G^h_{x,y}$ for $l = 1, 2, \ldots, g(h)$, where $g(h) = ah + b$ with $a, b \in \mathbb{N}$ and $a > 0$*

By assuming over the $g(h)$ terms, we can approximate the γ-decaying heuristic as follow.

$$\tilde{H}(x, y) := \eta \sum_{l=1}^{g(h)} \gamma^l f(x, y, l). \tag{3}$$

The approximation error is bounded as follows.

$|H(x, y) - \tilde{H}(x, y)| = \eta \sum_{l=g(h)+1}^{\infty} \gamma^l f(x, y, l) \leq \eta \sum_{l=ah+b+1}^{\infty} \gamma^l \lambda^l = \eta(\gamma\lambda)^{ah+b+1}(1 - \gamma\lambda)^{-1}$

Thus, $H(x, y)$ can be approximated from $G^h_{x,y}$, and the approximation error decreases at least exponentially with h.

In this work, we applied the SimRank score (Jeh and Widom, 2002), which is motivated by the notion that two nodes will be considered similar if their neighbors are similar. It is defined in the following recursive way: if $x = y$, then $s(x, y) := 1$; otherwise,

$$s(x, y) := \gamma \frac{\sum_{a \in \Gamma(x)} \sum_{b \in \Gamma(y)} s(a, b)}{|\Gamma(x)| \cdot |\Gamma(y)|}; \gamma \in [0, 1] \tag{4}$$

Once the link-based similarity is calculated, we apply the Binary Cross-entropy with logits as the loss function.

4.2. A novel approach to embed medical knowledge

We now describe our novel approach to embed medical knowledge from the 'observed sequence of procedures' into a stochastic policy the medical personnel follows.

We define a data structure we call the 'Observed Procedures Graph' (OPG). In the OPG, there is a single node for each observed action a, and there is an edge from a_1 to a_2 if there was one procedure P in which a_1 was executed immediately before a_2. Suppose the same transition is repeated, e.g., from a_1 to a_2. In that case, they are accumulated and documented as a single directed edge with a label representing the number of observed transitions. The OPG depicts the collected database and the 'observed sequences of procedures' via graph visualization (See Fig. 1).

It is vital to notice that although medical planning allows shifts in the chronological order of actions, the order is not sporadic. Since physician behavior is based on medical knowledge and clinical guidelines, we claim that a computer can find patterns and rules in physicians' actions. Therefore, we can exploit information in the OPG model to extract an implicit representation of these rules and medical knowledge. Nevertheless, the information presented in our OPG is sparse and often incomplete.

The phenomenon of incomplete knowledge affects prediction model accuracy in addition to having devastating results on the process of embedding knowledge (He et al., 2020). A recent paper by He et al. (2020) proposed harnessing the potential of graph networks to overcome this challenge. In our study, we applied similar methods, precisely the link prediction task, to find new patterns and sequences in the OPG. By doing so, we can enrich our database with a new'sequence of actions that have yet to be observed.

We start by embedding the graph. From the set of 'observed sequences of procedures', we omit a procedure P_i, and construct an OPG_i (OPG without the omitted procedure P_i) from the remaining procedures. Afterward, the OPG_i, still in its graph structure, is passed into a GCN, producing an embedded OPG_i graph. Here we use the 'node2vec' algorithm (Grover and Leskovec, 2016), which optimizes a neighborhood-preserving objective by simulating biased random walks from each node of the graph to sample directed acyclic sub-graphs. (See lines 1–12 in Algorithm 2 and Algorithm 1) This methodology balances the exploration–exploitation trade-off, which leads to representations obeying a spectrum of equivalences (Grover and Leskovec, 2016). The main advantage of graph embedding is to encode nodes into a latent vector space, which will later allow us to quantify node similarity. From this point forward, when we refer to the OPG model, it will be the embedded graph - OPG_E.

After obtaining the embedded OPG (OPG_E), we now proceed with the next step — train a link prediction model called the 'Implicit Procedures Graph' (IPG). In this process, we mask (remove label) a random node and all its edges from the previous omitted procedure P_i and pass it through the GCN to produce another graph embedding. The training objective is to correctly predict the label of the masked node and the masked edges, whether there is an edge between the masked node and any other node in P_i (See lines 13–19 in Algorithm 2). Specifically, the 'linking' process is based on a similarity score between nodes and edges embedding while using the Binary Cross-entropy to calculate the model loss. Such a training scheme makes this model bidirectional (See lines 13–19 in Algorithm 2).

The trained link predictor model allows us to find hidden relations and connections in the embedded OPG. The new information will enrich the representation and may compensate for the missing procedures.

The IPG model is the final step in embedding medical knowledge. Once the IPG is well trained, after a few modifications to the model we will elaborate on in the next paragraph, it can provide us with probabilistic predictions of the following anticipated action. To our knowledge, we are the first to use modern graph network techniques to embed medical knowledge from observations of clinical procedures.

Algorithm 1 GenGraph - Graph Generation Function

1: **Input:** $\{\pi_1, \ldots, \pi_n\}$ *#Set of observed execution traces*
2: **Output:** $\{V, E\}$
3: $V = \{a_i \mid \exists i, a_i \in \pi_i\}$ *#Set of vertices*
4: $E = \{\langle a, a', |\{i \mid \exists j, \pi_i[j] = a \cap \pi_i[j+1] = a'\}|\rangle, a, a' \in V$ *#Set of edges*

4.3. Action anticipation

Action anticipation refers to using a model to predict ahead of time, where the 'prediction horizon' defines the extent of the future predictions (Herath et al., 2017; Chandra et al., 2021).

Algorithm 2 IPG Construction - Pseudo Code

Input: $H = \{\pi_1, \ldots, \pi_n\}$,
nn.GCN *#Graph convolution neural network*
Output: IPG
GCN.Initialize = True *#Initialize model weights*
GCN.Task = LP *#Link Prediction*
GCN.Loss = BCEWithLogits *#Binary Cross Entropy with Logits*
N = NumEpochs *#Number of epochs*
$n = |H|$ *#Number of procedures in H*
for $i = 1$ **to** N **do**
  for $k = 1$ **to** n **do**
    $OPG^k = GenGraph(H \setminus \{\pi_k\})$
    $OPG_E^k = node2vec(OPG^k)$
    $G^k = GenGraph(\{\pi_k\})$
    $j = rand(1, \ldots, |\{\pi_k\}|)$
    $G^{k,j} = G^k$
    $G^{k,j}.V[j] = unk$ *#Replace label to "unknown"*
    $G^{k,j}.E[\langle a, a' \rangle \mid a \cup a' = V[j]] = NaN$
    $G_E^{k,j} = node2vec(G^{k,j})$
    $IPG = GCN.Train(OPG_E^k, G_E^{k,j}, G^{k,j})$
  end for
end for

To achieve this, we fine-tune the link-prediction model described above on a slightly different task – to predict the following action. Precisely, for the IPG model to predict an action, we modified it by adding a softmax activation layer at the end of the network and applying argmax to choose the most probable action to make a prediction. Now, to predict more than a single action (one step ahead), we apply this predictor recursively. We assume the most likely action in step $k-1$ was correctly predicted and use the network to predict the k–th action. The loss of each prediction was calculated using the MSE between the probability of the prediction and the ground truth action. The total loss was averaged based on the length of the procedure. Note that this task is not bidirectional, unlike the link prediction task.

To evaluate the execution of action anticipation, we can address the task as a multi-class classification and assess our model performance using the F_1-score (Dong et al., 2022).

4.4. Implementation details

As mentioned earlier, each simulation checklist report was generated as a chronologically ordered sequence of clinical actions in the form of a table. We then transform the table into a procedure using 'NetworkX' (Hagberg et al., 2008), and by joining all of the procedures together, we form a graph. This is the foundation for the framework OPG.

The embedding of medical knowledge was achieved by using a GCN for graph representation and Link Prediction for finding new connections and patterns in the directed graph. The network architecture is as follows: first, the input OPG graph is passed into a graph embedding layer. We concatenate the various contextual attributes for each entity in the graph to obtain their embeddings. These embeddings form an initial feature vector of entities to be used in the training. Afterward, we apply six layers of graph convolutions and batch normalization followed by the Sigmoid function. Now we can add new information to our model by training it with the link prediction task. As of now, the OPG model is referred to as IPG. We evaluate the IPG new connections under the closed-world assumption, in which unobserved connections between two nodes in a given executed procedure are false. This assumption transforms the evaluation into a well-defined task, as models are judged solely by their ability to fit known data. The loss function on the link prediction task is the binary cross-entropy with logits. Afterward, once the model was sufficiently trained, i.e., there was no evidence of improvement in the accuracy and loss values after several epochs, we evaluated the IPG model domain knowledge (embedded medical knowledge) by predicting the residents' future actions.

The computing infrastructure for running experiments included a single NVIDIA Tesla V100 GPU with 32 GB of memory and a Linux 20.04.2 LTS operating system. The proposed network was implemented using PyTorch 1.9. The dimension of contextual node embeddings was set to 1024. The network was trained from scratch for 120 epochs, while the fine-tuning steps were trained with an additional 30 epochs. We also experimented with other settings and found that small changes did not change the results much. Both training and fine-tuning stages were trained with a batch size of 1 using the binary cross-entropy with a logits loss function. The model parameters were trained with an ADAM optimizer with a learning rate of 0.0001 and 0.001 for training and fine-tuning steps, respectively. The best model parameters were selected based on the development set. In constructing our network architecture, we applied methods from 'NetworkX' - a Python language package for exploring and analyzing networks and network algorithms.

4.5. From action prediction to CDSS

Several medical actions, such as administering medication or defibrillator shocks, require time for preparation. In time-stressed situations, saving seconds from these procedures is vital. Thus, we develop a relatively simple CDSS that advises the medical personnel to prepare these instruments in advance. The framework predicts that one of these actions will occur in one of the next k steps with a probability greater than some threshold τ.

We now describe three aspects of CDSS to which we assessed our method: defibrillator management, medication management, and time-saving.

4.5.1. Defibrillator management

To validate our results, we calculated the accuracy of the defibrillator usage — the system should never predict a defibrillator in the anaphylaxis scenario. It should predict it at least once for severe bradycardia.

4.5.2. Medication management

Adrenaline injection plays a crucial role in the successful management of resuscitation. Based on the Advanced Cardiovascular Life Support (ACLS) algorithms, the patient is expected to be treated with several adrenaline injections at constant intervals during resuscitation.

We examined the number of times the system would have suggested the adrenaline injection during resuscitation based on the resident actions during this time interval. We compared this number and the number of times the examinee requested an adrenaline injection with the required amount (based on guidelines). We evaluated the results using the interclass correlation coefficient (ICC) (Koo and Li, 2016). This descriptive statistic assesses the consistency of the quantitative measurements made by different observers measuring the same quantity. An ICC score lower than 0.5 is considered poor reliability, 0.5 – 0.75 is considered moderate reliability, 0.75 – 0.9 is good reliability, and greater than 0.9 is excellent reliability. We calculated the ICC score based on the One-way random effects formula:

$$\frac{MS_R - MS_E}{MS_R} \tag{5}$$

Where MS_R is the mean square of each rater and MS_E is the mean square for error.

For this work, we applied this metric to assess the level of agreement between the recommended guidelines and the framework, and the resident performance.

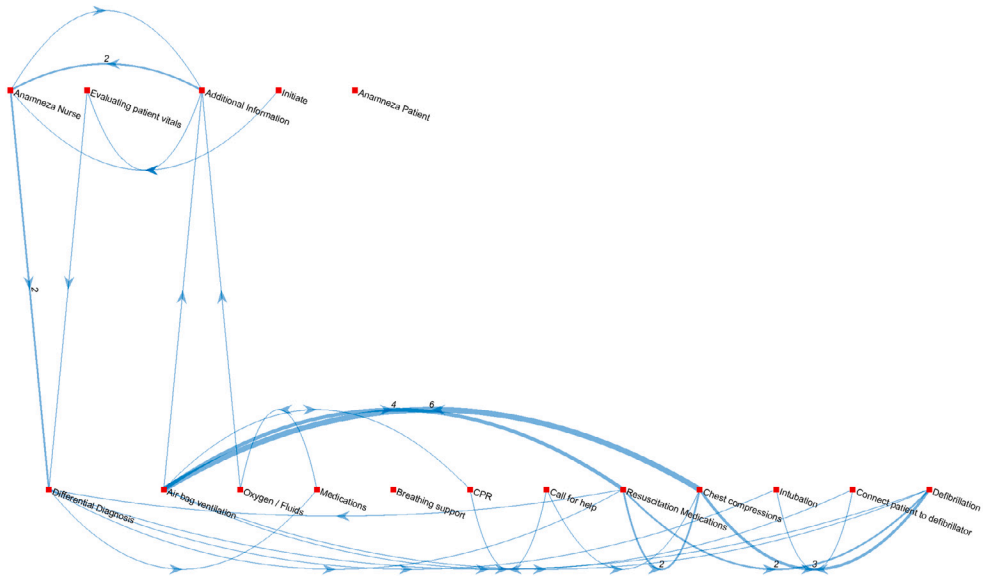

Fig. 1. VF observed sequences of procedures. Each vertex, observed action, represents a different task from the checklist, and the width of each edge represents the number of observed transitions. The appearing value indicates the most repeated transition. The arrow on the edges indicates their chronological order.

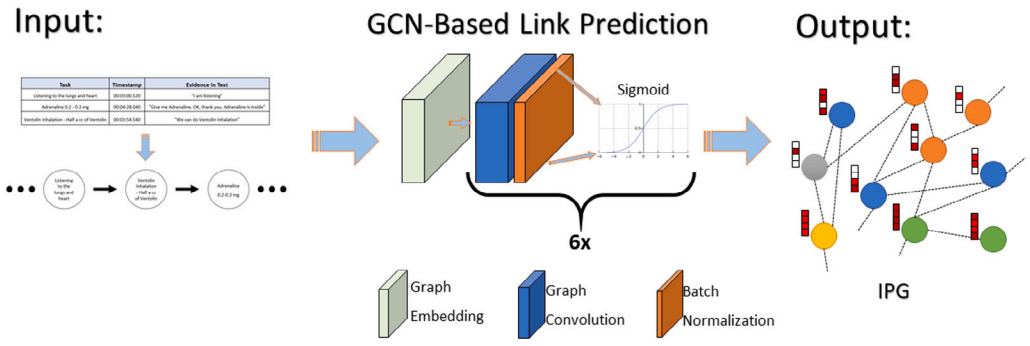

Fig. 2. Illustration of the proposed pipeline for embedding medical knowledge via GCN.

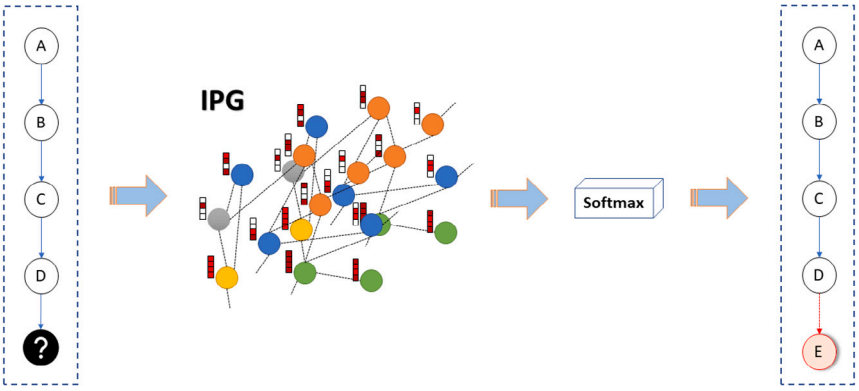

Fig. 3. Illustration of the proposed graph-based CDSS.

4.5.3. Time-saving

Another aspect of the CDSS we assessed is the time that could be saved by using our system. We predicted the following two actions: adrenaline injection and defibrillator usage. The "horizon" for both actions is $k = 1, 2$. When the system has a high enough confidence, $\tau = 0.75$, about one of the actions, it provokes a suggestion. We also used a 3-fold cross-validation approach to validate the following results.

5. Results

5.1. Evaluation of action prediction accuracy

Our action anticipation system was evaluated based on a prediction of one and two steps ahead. Both TAS datasets were divided to train-validation-test sets (70%:15%:15% ratio, respectively), and a 10-fold

Table 2
Model evaluation results on temporal action segmentation and medical simulations datasets.

Dataset	Task	F_1-score
50Salads (Stein and McKenna, 2013)	One-step prediction	0.611
	Two-step prediction	0.551
Assembly101 (Sener et al., 2022)	One-step prediction	0.690
	Two-step prediction	0.627
Medical Simulations	One-step prediction	0.899
	Two-step prediction	0.714

Table 3
Model prediction results for adrenaline injection and defibrillator usage. The time format is MM:SS ± Standard Deviation.

Action	One step	Two steps
Adrenaline	$00:38 \pm 00:12$	$01:34 \pm 01:07$
Defibrillator	$00:53 \pm 00:44$	$01:22 \pm 01:06$

cross-validation approach was applied to validate the reported results. Likewise, the medical dataset was divided to train-validation-test sets (80%:10%:10% ratio, respectively), and a 5-fold cross-validation approach was applied to validate the results in Table 2.

Ke et al.'s work (Ke et al., 2019) is the only published paper that evaluated action anticipation on the 50Salads datasets. Although their results are superior to ours, 65% and 61%, respectively, they used the video sequences and the labels. As for Assembly101, to our knowledge, we are the first to present results for the action anticipation task. We also performed an ablation study to determine the contribution of each network component to the overall system, which is included in the appendix.

5.2. Evaluation of CDSS

5.2.1. Defibrillator management accuracy

In 3 of the 31 bradycardia simulations, the resident failed to request the defibrillator, while the system successfully predicted the need for a defibrillator in all 31 simulations. This shows the potential of our system to serve as a decision-support system in a real clinical environment. Out of 31 anaphylaxis simulations, our system failed to predict the correct treatment in just two simulations and suggested using a defibrillator. Based on these results, our system can correctly predict the use/misuse of the defibrillator.

5.2.2. Medication management accuracy

As mentioned in Section 4.5.2 ("Medication Management"), we examined the number of times the system would have suggested the adrenaline injection during resuscitation based on the resident actions during this time interval. We compared this number with the expected amount, which is based on clinical guidelines (i.e., ground truth). We execute the same procedure for the number of times the examinee requested an adrenaline injection. The system has outmatched the resident with an ICC of 0.871, while the human participants achieved an ICC score of 0.510. This means the system can correctly identify the need for an adrenaline injection. This implies it can identify and follow ACLS protocols, even in real-life settings.

5.2.3. Time-saving evaluation

Table 3 shows the average time saved by using our system, the average time between when our system predicted one of these actions and when the resident performed it. As the results indicate, we can predict actions almost a minute before executing them accurately.

6. Discussion

Most of today's CDSS frameworks are based on knowledge graphs (KG), where the knowledge is stored in a highly expressive fashion and implicit hierarchical construction. This makes KG beneficial to many real-world settings, including the medical domain. Nevertheless, these systems cannot interpret and reason with clinicians' workflow; therefore, they are unsuitable for real-time medical assistance.

In this study, we described a technique to overcome this challenge by embedding medical knowledge from observations of procedures. We utilized a GCN to generalize from a few observed procedures to construct an 'Observed Procedures Graph' (OPG). Afterward, a 'Link Prediction' technique allowed us to find new procedure patterns and add them to the model. We referred to it as an 'Implicit Procedures Graph' (IPG). We investigated our proposed method on benchmarks of procedural actions originating from different domains to assess its robustness. In addition, we also assessed the framework results on real clinical datasets. The empirical results indicate that observations of action sequences are a suitable alternative for embedding medical expert knowledge. In addition, the proposed methodology is a suitable solution to overcome data limitations.

We then examined the applicability of combining our methodology with the speech-based action recognition system to construct a CDSS suitable for real-life medical assistance. This is considered a challenging objective because of the required level of awareness and understanding expected of the system, which will not interfere with medical personnel's work except when needed. In addition, during medical emergencies, CDSS is expected to encounter ambiguous information and numerous disruptions in the work environments. These factors combined may affect the performance of the CDSS.

The combined system is entirely autonomous, and a fully automatic pipeline from raw audio files to a CDSS was established. Since the system only requires audio signals, it does not interfere with the medical personnel workflow and minimizes the invasion of privacy.

We first tested our system's ability to predict the next step in the procedure. We then evaluated the system's applicability using two practical applications: medication management and real-time intervention. Both evaluations have shown promising results on the collected simulation data. Therefore, our framework has proven its potential to supervise the activity during a medical emergency and to assist in a complex decision-making situation.

Yet, the method we used for action recognition relies solely on keywords in the participants' speech, therefore, has limited accuracy. A significantly more extensive database is required to develop a more complex algorithm.

Another limitation we need to consider is the lack of sufficient participants. By exposing the model to a broader range of physicians' behavior, we can improve its predictions and certainty. We continuously collect data from many participants and a more comprehensive range of medical scenarios. This will facilitate the development of more complex frameworks, enhancing the CDSS functionality.

Medication dosage, video recording, and temporal information can provide the model with additional information which produces better predictions. Another aspect to consider is how these systems can be applied in the hospital and assist the medical personnel in working without disturbing the workflow.

7. Ethical considerations

First, we assert that all procedures contributing to this work comply with the ethical standards of the national and institutional committees on human experimentation and with the Helsinki Declaration.

Second, the CDSS we describe only offers support and suggestions to the medical personnel, which they can easily accept or ignore. This approach allows physicians to include our advanced CDSS as a component of their decisions while maintaining professional autonomy.

Thus, the medical personnel is still accountable for any action executed during treatment. In addition, any procedure that contains harmful actions is removed from our database. It is important to note that our CDSS was not used to treat real patients but was evaluated offline based on data from medical simulations. Thus, no patients were involved in either data collection or the CDSS evaluation.

8. Conclusion

We developed a CDSS framework that combines medical knowledge obtained by observing medical procedures with sequential information. Our model can generalize the procedure and find new possible combinations by learning a given sequence's unique structure and characteristics. This novel approach can leverage information and observations otherwise unobtainable by humans. Moreover, based on medical knowledge and clinical practice guidelines, our model can generate new sequences that human physicians have yet to perform.

In this work, we present a completely autonomous CDSS framework. In future work, we will examine potential downfalls, interface design, and the mental model of human users. We should keep in mind that judging whether a physician should accept the suggestion is an additional burden in an intense setting. In addition, further research will also address the implication aspects in real-life hospital settings and the effect of such framework on healthcare professionals' workflow.

CRediT authorship contribution statement

Sapir Gershov: Methodology, Software, Validation, Formal analysis, Investigation, Data curation, Writing – original draft, Visualization. **Aeyal Raz:** Conceptualization, Investigation, Resources, Data curation, Supervision, Project administration, Funding acquisition. **Erez Karpas:** Methodology, Investigation, Resources, Data curation, Project administration. **Shlomi Laufer:** Software, Validation, Resources, Writing – review & editing, Supervision.

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.

Data availability

Data will be made available on request.

Acknowledgment

This research was supported by VATAT Fund to the Technion Artificial Intelligence Hub (Tech.AI).

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.engappai.2023.107215.

References

Abdel-Basset, M., Manogaran, G., Gamal, A., Chang, V., 2020. A novel intelligent medical decision support model based on soft computing and IoT. IEEE Internet Things J. 7 (5), 4160–4170. http://dx.doi.org/10.1109/JIOT.2019.2931647.

Banning, M., 2008. A review of clinical decision making: models and current research. J. Clin. Nurs. 17 (2), 187–195.

Begoli, E., Bhattacharya, T., Kusnezov, D., 2019. The need for uncertainty quantification in machine-assisted medical decision making. Nat. Mach. Intell. 1 (1), 20–23.

Challen, R., Denny, J., Pitt, M., Gompels, L., Edwards, T., Tsaneva-Atanasova, K., 2019. Artificial intelligence, bias and clinical safety. BMJ Qual. Saf. 28 (3), 231–237.

Chandra, R., Goyal, S., Gupta, R., 2021. Evaluation of deep learning models for multi-step ahead time series prediction. IEEE Access 9, 83105–83123.

Dong, J., Huo, Q., Ferrari, S., 2022. A holistic approach for role inference and action anticipation in human teams. ACM Trans. Intell. Syst. Technol. 13 (6), 1–24.

Ernst, P., Siu, A., Weikum, G., 2015. Knowlife: a versatile approach for constructing a large knowledge graph for biomedical sciences. BMC Bioinform. 16, 1–13.

Fitriyani, N.L., Syafrudin, M., Alfian, G., Rhee, J., 2020. HDPM: An effective heart disease prediction model for a clinical decision support system. IEEE Access 8, 133034–133050. http://dx.doi.org/10.1109/ACCESS.2020.3010511.

Gaba, D.M., Howard, S.K., Fish, K.J., Smith, B.E., Sowb, Y.A., 2001. Simulation-based training in anesthesia crisis resource management (ACRM): a decade of experience. Simul. Gaming 32 (2), 175–193.

Gershov, S., Braunold, D., Spektor, R., Ioscovich, A., Raz, A., Laufer, S., 2023. Automating medical simulations. J. Biomed. Inform. 104446.

Gershov, S., Ringel, Y., Dvir, E., Tsirilman, T., Zvi, E.B., Braun, S., Raz, A., Laufer, S., 2021. Automatic speech-based checklist for medical simulations. In: Proceedings of the Second Workshop on Natural Language Processing for Medical Conversations. pp. 30–34.

Giordano, C., Brennan, M., Mohamed, B., Rashidi, P., Modave, F., Tighe, P., 2021. Accessing artificial intelligence for clinical decision-making. Front. Digit. Health 3, 645232.

Gong, F., Wang, M., Wang, H., Wang, S., Liu, M., 2021. SMR: medical knowledge graph embedding for safe medicine recommendation. Big Data Res. 23, 100174.

Grover, A., Leskovec, J., 2016. node2vec: Scalable feature learning for networks. In: ACM SIGKDD 2016. pp. 855–864.

Hagberg, A., Swart, P., S. Chult, D., 2008. Exploring network structure, dynamics, and function using networkx. Technical Report, Los Alamos National Lab.(LANL), Los Alamos, NM (United States).

He, B., Zhou, D., Xiao, J., Jiang, X., Liu, Q., Yuan, N.J., Xu, T., 2020. BERT-MK: Integrating graph contextualized knowledge into pre-trained language models. In: Findings of the Association for Computational Linguistics: EMNLP 2020. Association for Computational Linguistics, Online, pp. 2281–2290. http://dx.doi.org/10.18653/v1/2020.findings-emnlp.207, URL https://aclanthology.org/2020.findings-emnlp.207.

Herath, S., Harandi, M., Porikli, F., 2017. Going deeper into action recognition: A survey. Image Vis. Comput. 60, 4–21.

Hettige, B., Wang, W., Li, Y.-F., Le, S., Buntine, W., 2020. MedGraph: structural and temporal representation learning of electronic medical records. In: European Conference on Artificial Intelligence 2020. IOS Press, pp. 1810–1817.

Jeh, G., Widom, J., 2002. Simrank: a measure of structural-context similarity. In: Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 538–543.

Kawamoto, K., Houlihan, C.A., Balas, E.A., Lobach, D.F., 2005. Improving clinical practice using clinical decision support systems: a systematic review of trials to identify features critical to success. Bmj 330 (7494), 765.

Ke, Q., Fritz, M., Schiele, B., 2019. Time-conditioned action anticipation in one shot. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9925–9934.

Kim, K., Yang, H., Yi, J., Son, H.-E., Ryu, J.-Y., Kim, Y.C., Jeong, J.C., Chin, H.J., Na, K.Y., Chae, D.-W., et al., 2021. Real-time clinical decision support based on recurrent neural networks for in-hospital acute kidney injury: External validation and model interpretation. J. Med. Int. Res. 23 (4), e24120.

Kipf, T.N., Welling, M., 2016. Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907.

Koo, T.K., Li, M.Y., 2016. A guideline of selecting and reporting intraclass correlation coefficients for reliability research. J. Chiropr. Med. 15 (2), 155–163. http://dx.doi.org/10.1016/j.jcm.2016.02.012.

Kwan, J.L., Lo, L., Ferguson, J., Goldberg, H., Diaz-Martinez, J.P., Tomlinson, G., Grimshaw, J.M., Shojania, K.G., 2020. Computerised clinical decision support systems and absolute improvements in care: meta-analysis of controlled clinical trials. Bmj 370.

Li, Z., Abu Farha, Y., Gall, J., 2021. Temporal action segmentation from timestamp supervision. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8365–8374.

Nemati, S., Ghassemi, M.M., Clifford, G.D., 2016. Optimal medication dosing from suboptimal clinical examples: A deep reinforcement learning approach. In: 2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC). IEEE, pp. 2978–2981.

Patel, V.L., Shortliffe, E.H., Stefanelli, M., Szolovits, P., Berthold, M.R., Bellazzi, R., Abu-Hanna, A., 2009. The coming of age of artificial intelligence in medicine. Artif. Intell. Med. 46 (1), 5–17.

Prasad, N., Cheng, L.-F., Chivers, C., Draugelis, M., Engelhardt, B.E., 2017. A reinforcement learning approach to weaning of mechanical ventilation in intensive care units. arXiv preprint arXiv:1704.06300.

Schlichtkrull, M., Kipf, T.N., Bloem, P., Van Den Berg, R., Titov, I., Welling, M., 2018. Modeling relational data with graph convolutional networks. In: The Semantic Web: 15th International Conference, ESWC 2018, Heraklion, Crete, Greece, June 3–7, 2018, Proceedings 15. Springer, pp. 593–607.

Sener, F., Chatterjee, D., Shelepov, D., He, K., Singhania, D., Wang, R., Yao, A., 2022. Assembly101: A large-scale multi-view video dataset for understanding procedural activities. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 21096–21106.

Singhania, D., Rahaman, R., Yao, A., 2022. Iterative contrast-classify for semi-supervised temporal action segmentation. In: Proceedings of the AAAI Conference on Artificial Intelligence. Vol. 36, pp. 2262–2270.

Stein, S., McKenna, S.J., 2013. Combining embedded accelerometers with computer vision for recognizing food preparation activities. In: Proceedings of the 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing. pp. 729–738.

Sutton, R.T., Pincock, D., Baumgart, D.C., Sadowski, D.C., Fedorak, R.N., Kroeker, K.I., 2020. An overview of clinical decision support systems: benefits, risks, and strategies for success. NPJ Digit. Med. 3 (1), 17.

Wu, F., Souza, A., Zhang, T., Fifty, C., Yu, T., Weinberger, K., 2019. Simplifying graph convolutional networks. In: International Conference on Machine Learning. PMLR, pp. 6861–6871.

Yang, Q., Steinfeld, A., Zimmerman, J., 2019. Unremarkable AI: Fitting intelligent decision support into critical, clinical decision-making processes. In: Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems. pp. 1–11.

Zhang, M., Chen, Y., 2018. Link prediction based on graph neural networks. In: Advances in Neural Information Processing Systems. Vol. 31.

Zhang, S., Tong, H., Xu, J., Maciejewski, R., 2019. Graph convolutional networks: a comprehensive review. Comput. Soc. Netw. 6 (1), 1–23.

Zhao, Y., Xiong, Y., Wang, L., Wu, Z., Tang, X., Lin, D., 2017. Temporal action detection with structured segment networks. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 2914–2923.