Title	Deep convolution neural network model to predict relapse in breast cancer
Author(s)	Jha, Alokkumar; Verma, Ghanshyam; Khan, Yasar; Mehmood, Qaiser; Rebholz-Schuhmann, Dietrich; Sahay, Ratnesh
Publication Date	2018-12-17
Publication Information	Jha, Alokkumar, Verma, Ghanshyam, Khan, Yasar, Mehmood, Qaiser, Rebholz-Schuhmann, Dietrich, & Sahay, Ratnesh. (2018). Deep convolution neural network model to predict relapse in breast cancer. Paper presented at the 17th IEEE International Conference on Machine Learning and Applications (ICMLA 2018), Orlando, Florida, USA, 17-20 December, doi: 10.1109/ICMLA.2018.00059
Publisher	IEEE
Link to publisher's version	https://dx.doi.org/10.1109/ICMLA.2018.00059
Item record	http://hdl.handle.net/10379/14955
DOI	http://dx.doi.org/10.1109/ICMLA.2018.00059

2018 17th IEEE International Conference on Machine Learning and Applications

Deep Convolution Neural Network Model to Predict Relapse in Breast Cancer

Alokkumar Jha*, Ghanshyam Verma*, Yasar Khan*, Qaiser Mehmood*, Dietrich Rebholz-Schuhmann* and Ratnesh Sahay*
* Insight Centre for Data Analytics, National University of Ireland Galway
Email: {alokkumar.jha, ghanshyam.verma,yasar.khan, qaiser.mehmood, rebholz, ratnesh.sahay}@insight-centre.org

***Abstract*—A mishap in anti-cancer drug distribution is critical in breast cancer patients due to poor prediction model to identify the treatment regime in ER+ve and ER-ve (Estrogen Receptor (ER)) patients. The traditional method for the prediction depends on the change in expression across the normal-disease pair. However, it certainly misses the multidimensional aspect and underlying cause of relapse, such as various mutations, drug dosage side effects, methylation, etc. In this paper, we have developed a multi-layer neural network model to classify multidimensional genomics data into their similar annotation group. Further, we used this multi-layer cancer genomics perceptron for annotating differentially expressed genes (DEGs) to predict relapse based on ER status in breast cancer. This approach provides multivariate identification of genes, not just by differential expression, but, cause-effect of disease status due to drug overdosage and genomics-driven drug balancing method. The multi-layered neural network model, where each layer defines the relationship of similar databases with multidimensional knowledge. We illustrate that the use of multilayer knowledge graph with gene expression data for training the deep convolution neural network stratify the patient relapse and drug dosage along with underlying molecular properties.**

***Index Terms*—Deep learning, Knowledge Graph, Breast cancer, Neural Network**

I. Introduction

Multidimensional sparse functional annotation databases in genomics with hundreds of variables such as gene, protein, mutation, pathways, and drugs are available. Usually, these databases are available with more than one choice of selection for each type of information and incomplete or redundant information. In these settings prediction of disease or its effects with one dimension such as gene expression are challenging to detect with predictive information. The genomics data is multi-dimensional and databases usually spread across multiple databases. Due to this, a particular type of data (e.g., Mutations) can have multiple sources and hence the integration of this data with gene expression for prediction becomes a challenging task. However as explained in [1] and Figure 1 each layer represents broadly drug, mutations, disease, pathway and side effects associated with human genes. All these layers are knowledge graphs abbreviated as (KG-1 to KG-5). Moreover, as explained in Figure 1 each layer contains a combination of 5 layers treated as sub knowledge graphs. In this paper, we used gene expression data in combination with 5-hidden layers and 5*5 hidden sub-layers for prediction of ER+ve and ER-ve breast cancer patients. It's clear from Figure 1 that GE data along with sub-layers added in propagation hence neural network(NN) worked as a classifier. One of the key issues with genomics data and knowledge graph is incomplete or redundant data. The drug and mutations databases sparsely integrated across various platforms. Hence vector representation and controlling the dimensionality of each layer of data propagation obtain different outcomes [2]. For example, a known breast cancer gene BRCA2 due to its higher frequency of mutation and expression layer can be essential for ER+ve and ER-ve. However, in the second hidden layer (KG 2), while annotating BRCA2 with CNV as shown in Figure 1, it annotated with more copies of CNV for ER+ve group. This way multi hidden layer propagation provides a well-connected prediction. This type of prediction will lead to better biomarker discovery than traditional gene expression (GE only) based biomarker predictor in breast cancer [3]. Further, usually genomic features stated as (GE1-GE5) usually being used as an annotation to understand the mechanism of disease after the prediction using gene expression and survival data. Due to lack of connection between gene expression data and annotation databases, the models trained only on gene expression data usually provides expression biased predictors (biomarkers), and it always misses some key genes involved in disease progression [4]. Comparison of neural network based multi-layer predictors provide biomarkers with better survival than just gene expression-based biomarkers. Another issue is that the current methods and algorithms for predicting biomarkers for breast cancer uses the Random forest, Elastic net, SVM and NaÃ¯ve Bayes Model. Due to varying sample size, nature of experiments, the platform of gene expression data and their training and testing performance usually have significant variance. Since ER+ve and ER-ve breast cancer separated due to some pathway alteration during the cancer progression and data split from a single source is one of the reasons for the increased bias and variance in trained classifiers. Hence intrinsic noise in the class with some instances with the same attributes may have different classes(ER+ve and ER-ve). This misclassification results in higher training error. Increase in all these high factors lead to increase in mean error in training data as mentioned in the equation below:

978-1-5386-6805-4/18/$31.00 ©2018 European Union
DOI 10.1109/ICMLA.2018.00059

$$E(MSE) = noise^2[Gene\ Expression\ Platfrom] + Bias^2[Similarity\ in\ ER\ status] + Varaince[Late\ Annotation] \quad (1)$$

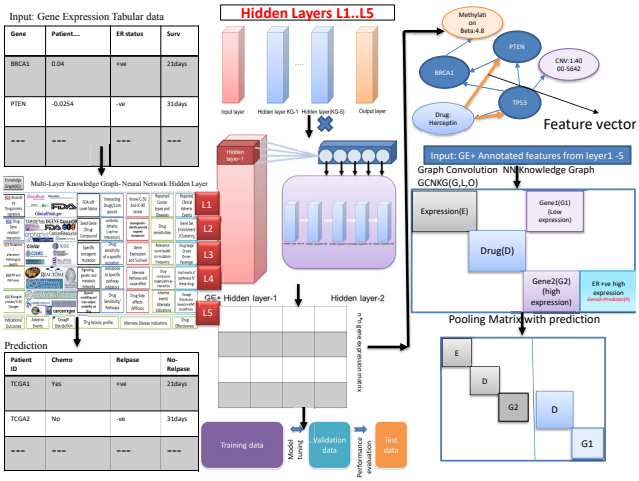

Fig. 1. Semantic linking of knowledge graph

In this paper, as shown in Figure 1, we have designed convolution knowledge graph neural network (CGCNN). The Input data for CKGNN is gene expression matrix where training matrix consists of expression value for each patient against 22,173 human genes. The training input data (GE matrix) will propagate through 5 hidden (GE-1 to GE-5) layers. Moreover, Within each layer, it will also add features from sub-layers of each layer. In the end, in pooling mode, it will provide a list of genes with prediction score to stratify ER+ve and ER-ve patients. Model and results discussed in upcoming sections.

II. Convolution NN-Layers and Knowledge Graph(KG)

Knowledge Graph(KG) is essential representation technique derived from graph-structured databases. However, its application in healthcare domain still seems to be far from reality. In this paper, we have represented a model for cancer genomics multi-dimensional data to extract novel biomarkers using knowledge graphs. One of the critical issues involved in biomarker discovery is entity resolution, where related entities are distributed in distinct databases either by similar or distinct identifiers or by the underlined domain related entity. The extraction from distinct knowledge bases contains clear information forms an intermediate knowledge discovery extraction graph. We have extended this process by single knowledge extraction graph for gynecological cancers $(OV, UCS, UCSC)$ and we refer to the task of removing noise, inferring missing information, and determining which candidate facts should include into a knowledge graph as knowledge graph identification. Cancer genomics data is an admixture of multidimensional datasets, and RDF representation of these data sets provides a unique relationship among these multidimensional entities. The example represents a unique relationship among disjoint datasets of Gene Expression (GE), Copy Number Variation(CNV) and Somatic Mutation datasets. All these datasets have sparsely distributed concerning various concepts. The traditional method of finding relationships among two domain related datasets is to derive linking properties, such as *owl:sameAs*. These techniques stand true when a person with domain knowledge can find parameters to link. However, there is a requirement of artificial intelligence to link these relations scientifically. There are two fundamental reasons behind that; 1. The data which is available in the form of knowledge graph is distributed among various repositories for each instance. 2. The data is continuously generating for knowledge enrichment in cancer genomics. This process of knowledge discovery and knowledge enrichment having three significant issues namely Entity Resolution, Node Labelling and Link Prediction and Ranking of the result. The advantage of knowledge graph for genomics data is for data integration to enrich functional annotation and data completeness. Completeness indeed is the core of knowledge graphs [7]. On the other hand rapidly growing RDF data in genomics, such as bio2rdf [24] and EBI-RDF [1] increase the demand of managing, mapping and integrating graph data more efficiently. One of best advantage with RDF representation of data (knowledge graphs) is that it can be queried using SPARQL [8]. As shown in Figure 1, five layers of knowledge graphs have been used as five hidden layers. All these layers have semantically linked with another layer. However, it is essential to have appropriate semantics for each layer. The conceptual interlinking of knowledge graph shown in Figure 2. Figure 2 explains the usability of KG-1 to KG-5 (Figure 1). For example, *BRCA1* gene used for input with gene expression values. As with each layer it adds CNV from *COSMIC*, pathway from *KEGG*, side effect from *SIDER*, ER status from *TCGA-clinical* and Methylated status from *TCGA*. It activated new link and relation among the entities across knowledge graphs from the same pair of gene and drug. Newly discovered links reveal the importance of having these KG hidden layers in the neural network.

The knowledge graph creation and its implementation with neural networks linked in further sections. Here Convolution of knowledge graphs helping to learn a function that can be applied for classification and regression of unknown links using hidden layers where two nodes of KG may not be in correspondence before KG creation [5]. Convolution Neural networks are known for sparse connectivity which exploits spatially-local correlation and local connectivity pattern between neurons of adjacent layers. Hence inputs of hidden units in the layer are from a subset of units sublayer, units that have spatially adjacent receptive fields. Convolution neural network is well-known for speech, text and image processing [6]. We have extended the well established CNN by combining it with Knowledge graphs for prediction of relapse in breast cancer with ER status and GE. We have also contributed RDF datasets

[1]https://www.ebi.ac.uk/rdf/

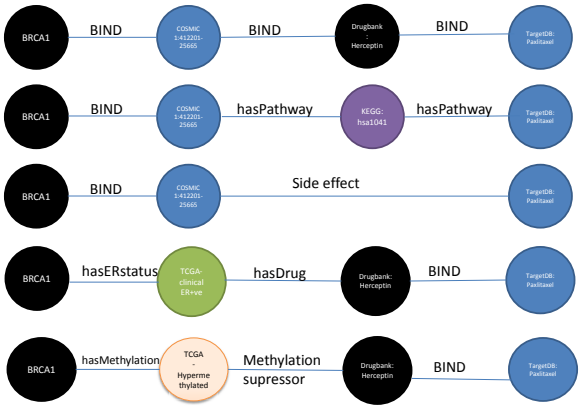

Fig. 2. The prediction model using graph neural network using multi-layer knowledge graph

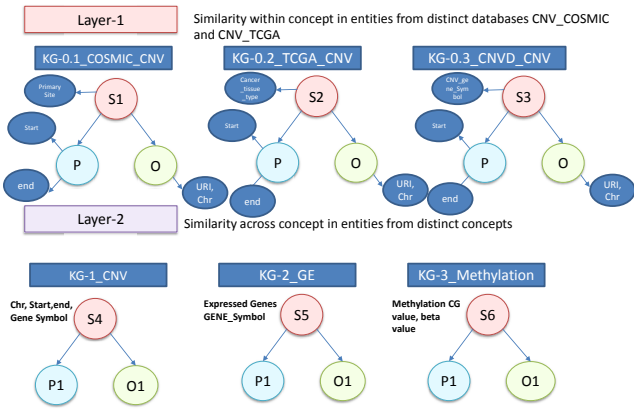

Fig. 3. prediction model using graph neural network using multi-layer knowledge graph

such as TCGA-OV, UCS, UCEC and CESC (Methylation, CNV and Gene expression in our previous work) [26], [27]

A. Knowledge Graph Creation

As formulated in [9], we created a master knowledge graph for different knowledge graphs with various categories based on entity similarity measure. In the paper, we are dealing with heterogeneous and multi-dimensional cancer genomics data. Definition of such KG explains directed graph $G = (V, E)$. As mentioned in Figure 3, the algorithm is taking *Subject(S)* as input from the $\mathcal{M}$ and mapping the subject against the subject of KG. In this process of mapping Predicate (P) and Object (O) may remain constant. As from Figure 3, lets assume that we are mapping Layer-2 1 with $\mathcal{M}$. Layer-2 has COSMIC and TCGA as databases. These databases have various types and genomic variants. These types and variants define the dimensionality of the knowledge graph. For instance as in Figure 3, we have selected gene expression (GE), Copy Number Variation (CNV) and DNA methylation (DM) selected as the first layer of mapping for $\mathcal{M}$ from KG. The first column of $\mathcal{M}$ has all Gene Symbols (GS), and these gene symbols can easily map with gene symbols of COSMIC GE. Since both the data COSMIC GE and $\mathcal{M}$ GE shares similar dimension, it is essential to define the priority annotation. To Solve this issue, we have extended the dimension towards COSMIC database. The another challenge here is mapping of single probe again multiple genes. Relation ship of many to many between genes and probes with added conjecture such as COSMIC GE and $\mathcal{M}$ increases learning depth. Here, gene with maximum mapping of probes are selected for learning. At KG layer genes can have high GE when comparing GE to GE, despite that if they have less mapped probes was not taken in account. This mapping method repeated with and across the KG layers. Once the choice becomes complex, then Algorithm Combined_Score (C) have been used to select the best annotation for all the G from $\mathcal{M}$. Enriched annotation $\mathcal{M}$ with multi-layer, multidimensional annotation and Combined_Score (C).

Definition 1: (**RDF Knowledge Graph for Convolution NN Hidden layers**). A knowledge graph is a directed graph $G = (V, E, \mathcal{R}, l)$, where V denotes vertices; E denotes the number of edges. R defines the predicates where $r \in \mathcal{R}$ and $l \subset \mathcal{R}$.

Definition 2: (**RDF Knowledge Graph for Convolution NN Hidden sublayers**). A sub knowledge graph is a directed graph $g = (v, e, \mathcal{R}, l)$, where v denotes vertices; e denotes the number of edges. R defines the predicates where $r \in \mathcal{R}$, $l \subset \mathcal{R}$, $v \subset V$, $e \subset E$. $g \in G$.

It is essential to understand the importance of these KG layers and the model learning using them. Since these five layers of KG have their own five sub-layers as shown in Figure 1, the formal definition of the KG Layer and Sub-layer is explained in **Definition 1 & 2.**

As explained in algorithm 1, once we have built the knowledge graph and sub knowledge graphs, the output will be gene expression matrix along with the annotation from five KGs as explained in Figure 1. The formal description for mapping of $\mathcal{M}$ demonstrated in Algorithm 1. As shown in Algorithm 1 each gene from $\mathcal{M}$ is beging extracted and then assigned all five leayers (L1 to L5) of as mentioned in Figure 1. Then annotations from each layer was check again the duplicate and unique entries to remove redundancey. The algorithm 1 uses entity matching concept of [21].

Once we have built the knowledge graph, the next thing is to combine the genes with KG, and since similar genes can have multiple annotations, it is essential to prioritize the annotation in the training data to get better classifiers. To achieve this, we demonstrate below, the ranking mechanism of annotation works based on the Combined_Score which is average of the Path_Score and the Association_Score(). As we can see from the algorithm 1 that the knowledge graph building will start selecting any of the layers from one of

Algorithm 1 Knowldge graph creation for CNN

Input: A Matrix $\mathcal{M}(a,b)$, a set of RDF Graphs $G = (\mathcal{V}_i, \mathcal{E}_i, \mathcal{R}_i)$ ▷ a = number of columns and b = no of rows
Output: A Matrix $\mathcal{M}\prime(\mathcal{M}, \mathcal{V}_i)$

procedure KG_CREATION($\mathcal{M}, \mathcal{G}_i$) ▷ Function to build knowledge graph with input expression matrix and RDF Graphs
  for $b = 0 \rightarrow\ b = n$ **do** ▷ total no of rows
    $r \leftarrow b[n]$
    $r = b[0] + r$ ▷ total no of Gene from matrix
    for $\mathcal{V}_i = 0 \rightarrow\ \mathcal{V}_i = b$ **do** ▷ total no of rows
      if $r \in \mathcal{G}_i(\mathcal{V}_i)$ **then** ▷ find genes in Graph
        for $i = 0 \rightarrow\ n$ **do** $\mathcal{G}_i = \mathcal{G}$ ▷ Assign KG-1
$\mathcal{M} \leftarrow \mathcal{M}_T \mathcal{G}_V$
$\mathcal{M}\prime \leftarrow \mathcal{M}_T \mathcal{G}_E$
$\mathcal{M} \leftarrow \mathcal{M}\prime \mathcal{R}$
$\mathcal{G}^{-1} \leftarrow \mathcal{M}_T + \mathcal{M}_T \mathcal{M}_T$
$\mathcal{G}_i \leftarrow$
$\begin{bmatrix} \mathcal{G} & \mathcal{M}\prime & r \end{bmatrix}$
        for $\mathcal{G}_{i=0}$ *to* $\mathcal{G}_{i=n}$ **do** KG_CREATION()
        end for
      end for
    end if
  end for
end for
end procedure

five layers and Gene expression data with gene names in the first column and expression values in all other columns. After this step, the gene names mapped against the gene names of annotation databases of each layer and a new column added to the gene expression matrix. A similar method applied to each layer of KG within their databases, and the corresponding column added in gene expression matrix. Since these datasets are in a silo and can help in knowledge enrichment as well as in knowledge validation, it is essential to have the ranking to select an appropriate and most relevant annotation as input for a better learning curve. We have used Combined_Score which is calculated using Path_Score and Association_Score().

Definition 3: A GCNN knowledge graph is a directed graph $G = (V, E, \mathcal{R}, l)$, where V denotes vertices; E denotes the number of edges and G defines the Knowledge graph annotations. The path score can be calculated as averaged path length from V to G where average calculated by shortest path(V,G) and longest path(V,G).

Definition 4: (Association_Score()). A knowledge graph is a directed graph $G = (V, E, \mathcal{R}, l)$, where V denotes vertices; E denotes the number of edges. G defines the Knowledge graph annotations then association can be calculated as

$$\phi = \frac{(supp(R * G) - supp(R) * supp(G))}{\sqrt{1 - supp(R) * (G)}} \tag{2}$$

Definition 5: (Combined_Score). For a given input if more than once choice is available then preference over these choices can be defined using combined Score $f : P \times A \leftarrow [0, 1]$, heer P_i and A_i are group of preferences score for instances from a layer and adjacent labels scores from another layer.Later, P_i and A_i was combined and relabeled based on following cases: $f(P_i, Ai) = f(x) = \begin{cases} 1, & x \leq\ a_i, p_i \\ 0, & \text{otherwise} \end{cases}$

Algorithm 2 Rank_Annotation()

Input: A Matrix $\mathcal{M}\prime(a,b)$ **Output:** A Matrix $\mathcal{M}\prime(\mathcal{M}, \mathcal{V}_i, Score)$
Initialize Matrix
/// Select matrix with annotated KG
for $i = 0 \rightarrow\ i = 4$ **do**
  for $g_i \in \mathcal{M}$, **do**
Path_Score(P)
Association_Score(A)

    $g_i \leftarrow P$
    $g_i \leftarrow A$
  end for
Combined_Score(C, P, A)
$C \in \mathcal{M}$
$\mathcal{M}\prime(a, b, g_i, C)$ $\mathcal{M}\prime \in \mathcal{M}$
end for

Once we obtained the gene expression matrix $\mathcal{M}$ along with annotations from algorithm 1, then we calculated the combined score using Path association and Path Length as mentioned in Definition 3, 4 and 5. Moreover, lastly, as per Algorithm 3, we calculated the Combined_Score for all annotations and added them back to $\mathcal{M}$. This way $\mathcal{M}$ is now having gene expression with five layers of annotation (hidden layers) and their combined score which is the input for learning graph convolution neural network, as shown in next section.

B. Deep Learning and Mathematical annotation for Relapse Prediction

Convolution Neural Network(CNN) is a method to extract features from an image using moving window called receptor. The adaptation of moving window receptor for knowladge graphs the instances of databases are arranged as filed of image pixel to generate an arranged spatial order. Spatial order will similarly as in the case of a pixel to identify the best receptor from a given layer for a querying entity. Extracted entities can be mapped to each receptor $\mathcal{R}$. Now in databases, it is difficult to find an orderly behavior within the mappings. To overcome this issue, we have constructed KG of databases, where we have annotated a matrix $\mathcal{M}$ with an RDF graph G along with GE and combined_Score(C). In case of pixel convolution, neural network works either from left to right or right to left. In this implementation, we added the annotation with a combined score which is sorted from maximum to minimum and vice-versa. Once the Genes(G) from $\mathcal{M}$ are sorted based on Combined_Score(C), we then built a co-expression correlation network. By using this correlation network, we built a neighborhood path among the entities of the graph. Then each graph is assigned a

hidden layer with every step of learning. To achieve this we built GCNN_ROOT_NODE() function in algorithm 3 which is partially adapted from [5].

Algorithm 3 GCNN_ROOT_NODE()

Input: For a Given Graph $\mathcal{G}_i(\mathcal{V}, \mathcal{E}$, label to graph $\updownarrow$, CNN kernel f, features $\mathcal{W}$ and receptor size $\mathcal{R})$

$\mathcal{G}(\mathcal{V}_{sort})$ = Top $\mathcal{W}$ elements of $\mathcal{V}$ according to Combined_Score($\mathcal{L}$)
$i = 1$, $j = 1$
while $j \geq \mathcal{W}$ **do**
    if $i \leq \mod(\mathcal{V}_{sort})$ **then**
        $\mathcal{F} = DEFINE\_FILTER(\mathcal{V}_{sort}(i))$
    else
$\mathcal{F} = DEFINE\_FILTER(\mathcal{M}_i(\mathcal{G}_i))$
apply $\mathcal{F}$ to each input tunnel
$i = i + s$, $j = j + 1$
apply $\mathcal{F}$ to each sub input tunnel $\mathcal{G}$
$i = i + s_i$, $j = j + 1$
    end if
Return $\mathcal{M}$
end while

Once we have determined the unique mappings and order of nodes through correlation path from one KG to another KG, it works as a receptor in case of the neural network. Now it is essential that each layer based on length of the path between two entities weighted with Combined Score(C). This embedding of path essentially identifies the shortest and longest path between a gene(G) from $\mathcal{M}$ and annotation from KG. Once the length of the path has been determined, the embedding is performed based on overall path length score determined by Association_Score() from **Definition 5**. The formal method to do this is mentioned in algorithm 4. As mentioned in Algorithm 4 each node of $\mathcal{G}_i$ with receptor size $\mathcal{R}$ was embedded with other KGs.

Algorithm 4 EMBED_KG_ER()

Input: $\mathcal{G}_i$(V,E), vertex of $\mathcal{G}_i$V, receptor size $\mathcal{R}$
Output: Set of embedded filed $\mathcal{E}$ for vertex V

$\mathcal{E} = V$
$\mathcal{T} = (V, \mathcal{M}_i)$
while $E \leq \mathcal{K}, E \geq \mathcal{L}$ **do** and
$T > 0$, $T \subset \mathcal{M}_i$
$T = \bigcup_{v \in T} E_i(V, \mathcal{M}_i)$
$E = (E \bigcup L) \cap (E \bigcup \mathcal{M}_i)$
$\mathcal{M}_i \geq 0$
end while
Return E

The earlier methods of breast cancer prediction in genomics were solely based on GE where they take use of higher and lower expression genes then stratify the risk group based on survival days and ER status. These methods were used to find out top variables from predictors and the annotation of these variables was a manual process. We use extracted annotation data for instance pathways , genomic locations for training each hidden layer to achieve higher performance and to find better biomarkers. Since annotations are noisy depends on the therapeutic level of databases it is essential to filter them. The algorithm to filter this annotation demonstrated in Algorithm 5.

Algorithm 5 DEFINE_FILTER()

Input: For a Given Graph $\mathcal{G}_i\mathcal{V}, \mathcal{E}$, label to graph $\mathcal{L}$, receptor size $\mathcal{R}$, Combined Score $\mathcal{C}$

$\mathcal{E} = DEFINE\_FILTER(v, \mathcal{R})$
$\mathcal{G}_i clus = KG\_Cluster(V, \mathcal{M}_i, \mathcal{C}, \mathcal{L})$
Return $\mathcal{E}$

Algorithm 5 primarily selectes the entities from algorithm 4 based on highest combined score (C) per gene and ignores rest of the annotations. At last, it yields filtered $\mathcal{M}$. This new filtered $\mathcal{M}$ can be formally defined by **DEFINITION 6.**

Definition 6: For a Given matrix $\mathcal{M}$ output of neuron of row x, column y in the l^{th} convolution layer and k^{th} feature pattern for t hidden layers defined as:

$$\mathcal{O}^l k_{x,y} = \int_{i=0}^{i=4} \mathcal{M}^i tanh(\sum_{t=0}^{f-1}\sum_{r=0}^{k_h}\sum_{c=0}^{k_w} \mathcal{W}^l.k_{r,c}^{\mathcal{O}^l} - 1 .r_{(x+r,x+c)} + Bias(l,k)) \quad (3)$$

As per the definition above at each propagation layer addition to the learned parameter after propagating through each hidden layer was defined to obtain similarity between annotated entities as

$$\mathcal{O}^l k_{x,y} = \int_{i=0}^{i=4} \mathcal{M}^i tanh(\sum_{t=0}^{f-1}\sum_{r=0}^{k_h}\sum_{c=0}^{k_w} \mathcal{W}^l.k_{r,c}.\mathcal{O}^l - 1 .r_{(x+r,x+c)} + Bias(l,k)) \quad (4)$$

The **Definition** 6 can be applied to each propagation layer while learning. Learning at each propagation layer can be defined by the following formula [20]:

$$g_\theta(\Lambda) = \sum_{k=0}^{K-1} \theta_k \Lambda^k \quad (5)$$

Now the learning algorithm of GCNN can be formally defined as a partition of the neural network as clusters in Algorithm 6. It is essential to Cluster the $\mathcal{M}$ during learning since most of the learning algorithms are injective. Hence reusability of the previous layer becomes extremely difficult during leaning which causes the drop during the learning process. To lineate this drop, partition of $\mathcal{M}$ based on receptor $\mathcal{R}$ reduces the *propagation loss*. CNN with the cluster is formally defined in Algorithm 6.

The training algorithm is demonstrated below. It is abbreviated as Graph Convolution Neural Network (GCNN). Traditionally CNN is being used for image processing. However,

Algorithm 6 KG_Cluster()

Input: Matrix $\mathcal{M}$, $\mathcal{G}_i(V, E)$, receptor size $\mathcal{R}$, label l and Combined Score (C)
Output: Matrix with receptor field $\mathcal{M}(\mathcal{R}) \cup \mathcal{R} \in V$

$\mathcal{G}_i \rightarrow f : \mathcal{R} \rightarrow \mathcal{R}$
for (**do**$\mathcal{G}_i$, $i = 0$ to $i = 4$)
  if $C(\mathcal{G}_i) > C_{i+1}$ **then**
$\mathcal{M} = \mathcal{M}_{i+1}$
  else
$\mathcal{M} = \mathcal{M}_i$
  end if
$\forall C\mathcal{G}_i$ then
$\mathcal{G}_i = (\mathcal{G}_i - \mathcal{R})/\sqrt{V}$
$\mathcal{G}_i \rightarrow \mathcal{R}$
end for
$Return\ \mathcal{G}_i|\mathcal{M}|$

we have replaced the hidden layers with 5 KGs, and each layer added in backpropagation. While training this we will have gene expression values, survival days, ER status, and annotation from KG-1 to KG-5 along with combined score to identify the most optimal annotation for each gene as discussed in the previous section. Output for each neuron and each layer extended from Liu et al [19].

III. Result preamble: Breast cancer and GCNN

Breast cancer defined as hormone receptor called as ER+ve, ER-ve, and $HER2\pm$. The stratification of patients based on $ER\pm$ helps to design the chemotherapy drug dosage in the patients. We have applied GCNN() in breast cancer patients with $ER\pm$ status. It is critical to understand the role genes which are driving the tumor progression based on ER status [22] to understand the sensitivity of the therapy. In this paper, we have integrated KG with gene expression and ER status and predicted the relapse in the patients based on top 20 gene (Table III) ranked using *Gini*. The Performance of the Genes predicted from GCNN is being compared with RF (Random Forest-15000 Trees), SVM (Support Vector Machine), NN(Neural Network n=1000). Further these markers have been compared in terms of Cox-proportion hazard ratio (H.R.) - Defined lethality of gene and Mean Survival time with the top 20 genes from the Algorithms RF, NN, SVM, GCNN and four benchmark papers Aziz et al., Naderi et al., Bieche et al., Peters et al. . Since, the Genes retrieved from GCNN is performing best-concerning accuracy measured as Area Under the Curve (A.U.C. in %) along with significant P-value < 0.05, the detailed results discussed in further sections.

IV. Result and Discussion

A. Prognostic Validation of Top Variable : Performance Analysis

Firstly, we trained the GCNN on TCGA-BRCA data and further validated with GSE47561. The results from Training and validation can be seen in Table I. The AUC has been calculated using Sensitivity and Specificity from Confusion Matrix generated from these algorithms. As we can see from Table I, GCNN has 94% and 91.9% AUC for training and validation, respectively, which is outperforming NN, RF, and SVM. Top 20 variables from training is mentioned in Table III.

TABLE I
GCNN Training and Validation

Algorithm	Training TCGA-BRCA		Validation-GSE47561	
-	AUC	P-value	AUC	P-value
NN	86[CI 78-87.2]	0.04	81 [CI 82-89.1]	0.06
RF	78 [CI 89.4-79.4]	0.02	84[CI 78.2-87.4]	0.01
SVM	70[CI 68.54-74.36]	0.01	85 [CI 82.25-89.3]	0.021
GCNN	94 [CI 92.8-96.1]	0.031	91.9 [CI 89.23-94.8]	0.044

Once we have trained and validated the model, further, we have tested the performance of top 20 variables (Genes) as mentioned under GCNN Table III retrieved from GCNN training set. The performance of the model has been tested on six independent datasets, as mentioned in Table II. The performance of Top 20 gene signatures from GCNN compared with NN, RF, and SVM as mentioned in Table III. The performance of GCNN is above 90% and better than all the data sets accept GSE25055 where the performance of GCNN is almost similar to NN.

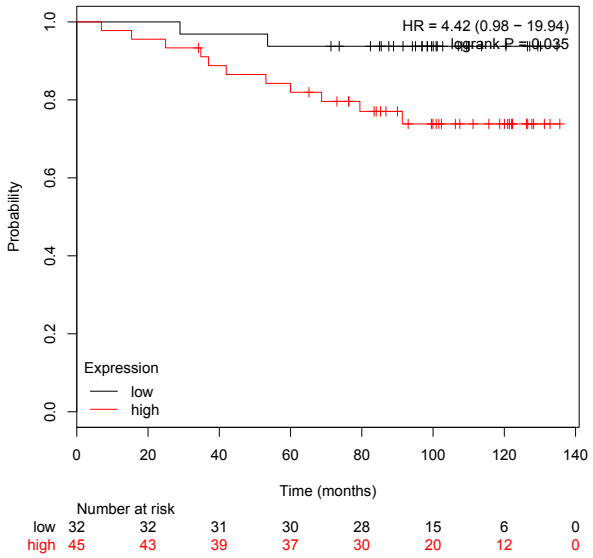

Fig. 4. Cox-Proportion Model for GCNN Genes

B. Diagnostic Validation: Comparison of Genes with other predictors and Survival Analysis)

Once we have tested the performance of the gene progostically through AUC, it is essential to see the diagnostic aspect of the gene apart from algorithmic performance to see if these

TABLE II
GCNN PERFORMANCE TABLE-TESTING. (*CONFIDENCE INTERVAL [CI])

Algorithm	GSE20685		GSE25055		GSE22219		GSE12276		GSE7390		GSE24450	
-	AUC	P-value	AUC	P-value	AUC	P-value	AUC	P-value	AUC	P-value	AUC	P-value
NN	91.9 [88.2-94.1]*	0.64	78 [64.8-81.3]*	0.01	66 [50.2-70.0]*	0.012	81 [77.4-84.25]*	0.021	69 [64-73.2]*	0.81	77 [68.7-81.7]*	0.04
RF	85 [83.7-91.3]*	0.032	59 [71.6-63.2]*	0.07	89 [78.9-89.5]*	0.03	88 [86.5-89.9]*	0.2	91 [81.8-91.4]*	0.3	66 [61.8-70.2]*	0.05
SVM	90.6 [86.9.2-93.01]*	0.041	77 [71.4-77.9]*	0.041	73 [67.5-74.6]*	0.04	77 [76-83.9]*	0.061	84 [82-89.1]*	0.032	71 [61.3-77.8]*	0.021
GCNN	91.7 [90.01-97.3]*	0.0231	94.5 [91-95.1]*	0.44	94.5 [93.2-94.8]*	0.001	91 [84.9-92.5]*	0.011	92.4 [87-95.1]]*	0.05	92.4 [90.1-98.1]*	0.029

TABLE III
COMPARISON OF GENES WITHIN ALGORITHMS AND OTHER PREDICTOR (H.R.=HAZARD RATIO, MST=MEDIAN SURVIVAL TIME)

Algo.	**GCNN**	**SVM**	**RF**	**CNN**	**Aziz** [18]	**Naderi** [15]	**Bieche** [16]	**Peters** [17]
Genes	ABCC5 DPP3 AP2S1 CDH11 TSPAN5 TSPAN1 SEC23B ARL4C RASGRP1 OPTN RAB11A MLPH ADCY3 ENPP1 GNB2 GNG4 SH3GLB1 COPE NFKB2 F3	UBD CEBPG AGR2 SLC19A2 IFI44 CXCL13 TACC2 DNPH1 ST6GALNAC2 TXNIP CENPF TMEM135 SLC1A4 UBE2Z C8orf33 SLC12A2 SLC25A1 SLCO2A1 SLPI SNAI2 SMARCA1 GRAMD3	CDH3 PDCD6 GNE ZNF587B TOM1L1 LOC100507577 ABCC5 DPP3 ATP9A CDH11 TSPAN5 TSPAN1 PPIF ARL4C RASGRP1 OPTN NAMPT AKAP9 LHFP MPHOSPH6	DNPH1 ST6GALNAC2 TXNIP CENPF FRMD6 SPG7 DUSP21 BRCA1 CD44 CCL19 CXCL11 CX3CL1 BIK SDC1 SDC2 SDC4 CXCL12 SDF2 NECAB3 SECTM1	MRPL52 TRIP13 ITPRIP SLC38A9 FRMD6 SORCS2 ELTDI NOTCH2 CPXCR1 OR10H5 PDC DUOX2 GFRA4 LASS6 OSBPL9 C12orf66 SPG7 DUSP21 BRCA1	DUFD1 ASPM SPAG5 FADD BAALC C10orf3 FLJ20641 BM039 MGC34923 KIAA0703 PSMD14 OMD A23P30055 EBP DCN EXO1 SHMT2 MELK FLJ14627 THC1964466 SHOX2	AR AREG ARHC/RhoC BCL2 BRCA1 BRCA2 CAV1 CCND1 CCNE1 CD44 CDH1 CGA CGB CP CXCL12 CXCR4 DNMT3B EGFR/ERBB1 ERBB2 ERBB3	TRIM44 SIRT2 C5AR1 PLK1 UBE2D2 NEU4 ADCY9 PAPSS2 HSS00095627 PNPLA2 LOC401021 ST3GAL4 CAMK1 VPS33A MS4A6A NOXA1 VPREB3 LOC253039 ITGB6 UNC93B1
H.R.	**4.42[CI 0.98-19.4]**	0.51[CI 0.17-1.52]	1.9[CI 0.37-3.24]	0.37[CI 0.1-1.34]	2.32[CI 0.78-6.93]	4.08[CI 0.53-31.39]	0.57[CI 0.19-1.7]	2.35[CI 0.72-7.63]
MST-Months	37	47	35	30	22	49	45	32
P-value	0.035	0.22	0.88	0.11	0.12	0.14	0.31	0.14

genes play any significant role into the patients stratified using GCNN. To Achieve this, we have Cox-Proportion Model to find the Hazard Ratio (H.R.) of these retrieved from GCNN and other algorithms. We have also tested GCNN genes with few published benchmarks. We have used KM-Plotter [23] for survival analysis.

All the H.R. has retrieved through GSE9195 data . The top 20 genes from each algorithm retrieved and benchmarked with H.R., MST, P-values are shown in Table III. As we can see GCNN genes have highest HR ration means a higher expression of this gene can affect the RPS (Relpase free survival) within significant p-value. However, survival time is better than all the algorithms and couple of benchmark datasets (85 days-approx). The survival curve for GCNN gene shown in Figure 4

As shown in Figure 4 GCNN genes have a confidence interval CI [0.98-19.4] shows the lethality of GCNN genes with only significant p-value-0.035 (criteria of significance p-value <0.05) in comparison with other predictor.

C. Empirical validation

We have empirically validated the model as mentioned in Figure 5. The residual learning method using convolution adds the feedback and improves the known usage of knowledge graph hence improves the performance. Here as mentioned in Figure 5, with traditional methods we can get maximum 100 annotations for 20 predictors. However using 5 -layer plain method with RF, NN we can get 28100 annotations. However,

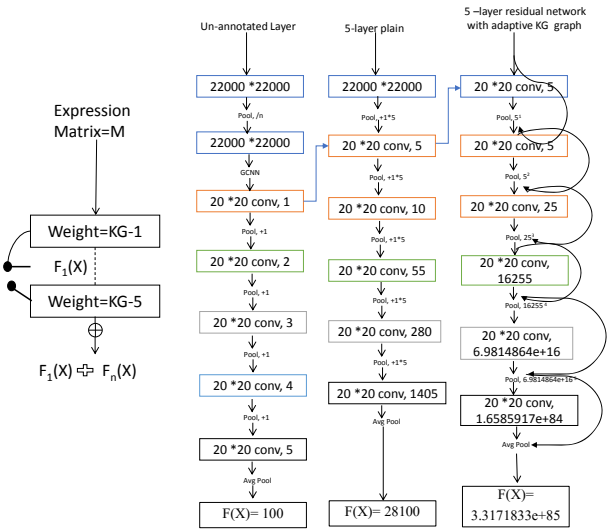

Fig. 5. Layered pooling and knowledge enrichment

with GCNN using 5-layer residual network with adaptive KG, we can get 3.3171833e+85 annotations for 20 genes. This way the 5 hidden layers designed as KG improves the prediction probability exponentially and hence improves the prediction.

V. Prediction Results and ER status

The genes retrieved from GCNN as shown in Table III. ER +ve breast cancer are essential types where cancer cell grows in response to the hormone estrogen. In the patients more towards ER -ve hormone therapy are more likely to work. ER -ve where no receptors are present the hormone therapy will not work. As mentioned in [25] that identifying the ER status is a daunting task. Our approach has been being able to predict the ER status in breast cancer patients. This way it will help in treatment planning in breast cancer. This approach can be used to build a general prediction model by reusing the features from our earlier compendium [28].

VI. Conclusion

In this paper, we have demonstrated 20 Gene signature to predict chances of relapse in Breast cancer (BRCA patients) using GCNN (Graph Convolution Neural network). Moreover, tested prognostic and diagnostic aspect of the gene against other existing algorithm and biomarkers and proved that GCNN genes are performing better. These genes can be used for drug dosage balancing in BRCA Patients apart from ER prediction.

VII. Acknowledgments

This publication has emanated from research conducted with the financial support of Science Foundation Ireland (SFI) under Grant Number SFI/12/RC/2289, co-funded by the European Regional Development Fund

References

[1] Jha, A., Mehdi, M., Khan, Y., Mehmood, Q., Rebholz-Schuhmann, D. and Sahay, R., 2016, September. Drug Dosage Balancing Using Large Scale Multi-omics Datasets. In VLDB Workshop on Data Management and Analytics for Medicine and Healthcare (pp. 81-100). Springer, Cham.

[2] Yang, B., Yih, W.T., He, X., Gao, J. and Deng, L., 2014. Embedding entities and relations for learning and inference in knowledge bases. arXiv preprint arXiv:1412.6575.

[3] Yousefi, S., Amrollahi, F., Amgad, M., Dong, C., Lewis, J.E., Song, C., Gutman, D.A., Halani, S.H., Vega, J.E.V., Brat, D.J. and Cooper, L.A., 2017. Predicting clinical outcomes from large scale cancer genomic profiles with deep survival models. Scientific Reports, 7(1), p.11707.

[4] Desmedt, C., Ruiz-Garcia, E. and André, F., 2008. Gene expression predictors in breast cancer: current status, limitations and perspectives. European journal of cancer, 44(18), pp.2714-2720.

[5] Niepert, M., Ahmed, M. and Kutzkov, K., 2016, June. Learning convolutional neural networks for graphs. In International conference on machine learning (pp. 2014-2023).

[6] Bhandare, A., Bhide, M., Gokhale, P. and Chandavarkar, R., 2016. Applications of Convolutional Neural Networks. International Journal of Computer Science and Information Technologies, pp.2206-2215.

[7] Pershina, M., Yakout, M. and Chakrabarti, K., 2015, October. Holistic entity matching across knowledge graphs. In Big Data (Big Data), 2015 IEEE International Conference on (pp. 1585-1590). IEEE.

[8] Zou, L., Mo, J., Chen, L., Özsu, M.T. and Zhao, D., 2011. gStore: answering SPARQL queries via subgraph matching. Proceedings of the VLDB Endowment, 4(8), pp.482-493.

[9] Choudhury, A., Sharma, S., Mitra, P., Sebastian, C., Naidu, S.S. and Chelliah, M., 2015, March. SimCat: an entity similarity measure for heterogeneous knowledge graph with categories. In Proceedings of the Second ACM IKDD Conference on Data Sciences (pp. 112-113). ACM.

[10] Roos, L., van Dongen, J., Bell, C.G., Burri, A., Deloukas, P., Boomsma, D.I., Spector, T.D. and Bell, J.T., 2016. Integrative DNA methylome analysis of pan-cancer biomarkers in cancer discordant monozygotic twin-pairs. Clinical epigenetics, 8(1), p.7.

[11] Cohen, W.W. and Sarawagi, S., 2004, August. Exploiting dictionaries in named entity extraction: combining semi-markov extraction processes and data integration methods. In Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 89-98). ACM.

[12] Martinez-Ledesma, E., Verhaak, R.G. and Treviño, V., 2015. Identification of a multi-cancer gene expression biomarker for cancer clinical outcomes using a network-based algorithm. Scientific reports, 5, p.11966.

[13] McCusker, J.P., Dumontier, M., Yan, R., He, S., Dordick, J.S. and McGuinness, D.L., 2017. Finding melanoma drugs through a probabilistic knowledge graph. PeerJ Computer Science, 3, p.e106.

[14] Venet, D., Dumont, J.E. and Detours, V., 2011. Most random gene expression signatures are significantly associated with breast cancer outcome. PLoS computational biology, 7(10), p.e1002240.

[15] Naderi, A., Teschendorff, A.E., Barbosa-Morais, N.L., Pinder, S.E., Green, A.R., Powe, D.G., Robertson, J.F.R., Aparicio, S., Ellis, I.O., Brenton, J.D. and Caldas, C., 2007. A gene-expression signature to predict survival in breast cancer across independent data sets. Oncogene, 26(10), p.1507.

[16] Bièche, I., Tozlu, S., Girault, I. and Lidereau, R., 2004. Identification of a three-gene expression signature of poor-prognosis breast carcinoma. Molecular cancer, 3(1), p.37.

[17] Peters, C.J., Rees, J.R., Hardwick, R.H., Hardwick, J.S., Vowler, S.L., Ong, C.A.J., Zhang, C., Save, V., O'Donovan, M., Rassl, D. and Alderson, D., 2010. A 4-gene signature predicts survival of patients with resected adenocarcinoma of the esophagus, junction, and gastric cardia. Gastroenterology, 139(6), pp.1995-2004.

[18] Aziz, N.A.A., Mokhtar, N.M., Harun, R., Mollah, M.M.H., Rose, I.M., Sagap, I., Tamil, A.M., Ngah, W.Z.W. and Jamal, R., 2016. A 19-Gene expression signature as a predictor of survival in colorectal cancer. BMC medical genomics, 9(1), p.58.

[19] Liu, T., Fang, S., Zhao, Y., Wang, P. and Zhang, J., 2015. Implementation of training convolutional neural networks. arXiv preprint arXiv:1506.01195.

[20] Defferrard, M., Bresson, X. and Vandergheynst, P., 2016. Convolutional neural networks on graphs with fast localized spectral filtering. In Advances in Neural Information Processing Systems (pp. 3844-3852).

[21] Zheng, W., Zou, L., Peng, W., Yan, X., Song, S. and Zhao, D., 2016. Semantic SPARQL similarity search over RDF knowledge graphs. Proceedings of the VLDB Endowment, 9(11), pp.840-851.

[22] Lønning, P.E., Knappskog, S., Staalesen, V., Chrisanthar, R. and Lillehaug, J.R., 2007. Breast cancer prognostication and prediction in the postgenomic era. Annals of oncology, 18(8), pp.1293-1306.

[23] Győrffy, B., Lanczky, A., Eklund, A.C., Denkert, C., Budczies, J., Li, Q. and Szallasi, Z., 2010. An online survival analysis tool to rapidly assess the effect of 22,277 genes on breast cancer prognosis using microarray data of 1,809 patients. Breast cancer research and treatment, 123(3), pp.725-731.

[24] Belleau, F., Nolin, M.A., Tourigny, N., Rigault, P. and Morissette, J., 2008. Bio2RDF: towards a mashup to build bioinformatics knowledge systems. Journal of biomedical informatics, 41(5), pp.706-716.

[25] Welboren, W. J., Stunnenberg, H. G., Sweep, F. C., & Span, P. N. (2007). Identifying estrogen receptor target genes. Molecular oncology, 1(2), 138-143.

[26] Jha, A., Khan, Y., Mehdi, M., Karim, M. R., Mehmood, Q., Zappa, A., ... & Sahay, R. (2017). Towards precision medicine: discovering novel gynecological cancer biomarkers and pathways using linked data. Journal of biomedical semantics, 8(1), 40.

[27] Jha, A., Khan, Y., Mehdi, M., Karim, M. R., Mehmood, Q., Zappa, A., ... & Sahay, R. (2017). Towards precision medicine: discovering novel gynecological cancer biomarkers and pathways using linked data. Journal of biomedical semantics, 8(1), 40.

[28] Jha, A., Khare, A., & Singh, R. (2015). Features' compendium for machine learning in NGS data Analysis.