2021 Second International Conference on Intelligent Data Science Technologies and Applications (IDSTA)

Material and Cost estimation of a Customized Product based on the Customer's description

Kevin Baboolal and Patrick Hosein
The University of the West Indies, St. Augustine, Trinidad
k.baboolal1.0@gmail.com, patrick.hosein@sta.uwi.edu

***Abstract*—Many companies develop customized products for their customers. The customer, with the assistance of the salesperson, will typically provide a description of the product and the company must then use that description to determine the type and quantities of materials required to produce the product. This information is then used to derive the cost of production from which a price can be determined. We use Machine Learning techniques to automate this process. The product description is analyzed using Natural Language Processing to extract the relevant information. This information, along with other attributes, are then fed into a Deep Neural Network (DNN). The DNN has an output for each possible component of the product with the output value equal to the quantity of that component required for the product. We illustrate the approach with a dataset taken from a company that builds electrical distribution boards. Each distribution board must be customized for the customer and so the accurate determination of the components and their quantities is vital in determining an appropriate price. Note that, the component list (called the Bill of Materials or BOM) also helps determine the processing required and this too affects the production cost. We illustrate the effectiveness of our approach with the data obtained from this factory.**

***Index Terms*—Machine Learning, Bill of Materials, Pricing Optimization**

I. Introduction

The engineering bill of materials (BOM) is a document instructing users on the components and quantities required for the manufacture of a product. The BOM document is primarily used across all manufacturing industries. Variants of the document exists in other industries such as civil construction, where it is referred to as a bill of quantities. In all instances the BOM reflects the product as designed by engineering and serves as a material requirement list for manufacturing. BOMs can exist in a multilevel tree structure with lower branches representing separate components that often undergo sub-assembly operations before inclusion into the final product [1]. The BOMs collected for this study do not have a tree structure but can be represented as a two dimensional array (Component Name, Quantity) for a given product. The structure is an important distinction for the method proposed as different techniques such as random forest regression may yield better results on a tree structured BOM. The inventory management department of a manufacturing company uses the BOM for the procurement of components and the issuing of materials to manufacturing. Our case study focuses on a class of manufacturing entities known as job shops. Jobs shops are characterized by high product variance and low production volume, the value proposition of job shop manufacturing is to tailor the product to the needs of the customer [2]. The order processing cycle begins with the receipt of a customer request. The request is answered with a quotation and, if the customer confirms the purchase, the order moves to the BOM estimation phase. BOM estimates are executed by a separate department that rely on historical product BOMs, approximate hand calculations and ad-hoc meetings with the engineering department to create the estimated BOMs. These estimates are used by the procurement department to order materials in advance of the detailed engineering being completed thereby reducing the overall delivery time. A just-in-time manufacturing system as described previously allows for reduced inventory carrying costs and higher profitability for companies. For this industry, each customer order requires a detailed engineering model to determine the materials and quantities required. Models are created using computer aided design software and can be a time consuming process. During periods of high volume, the business can lose valuable time as estimators may take weeks to complete large orders and designers may take even longer for detailed product drawings.

In this paper, we present the use of deep neural networks to predict BOMs for new products given the product quotation. In our approach we do not attempt to replace the human estimators however we seek to augment their work with machine learning to boost their productivity. In the next section we discuss related research in this area. In section III we describe the dataset being used for our experimentation and model evaluation. Section IV details the model and metrics to be used followed by an analysis of the results in section V. We then make our closing remarks and provide potential future work.

II. Related Work

Significant research has been done on an optimization problem often referred to as the "job shop problem". The simplest iteration of the problem describes the minimization of total production time for jobs that must be scheduled on machines of varying capacity. Often, these manufacturing constraints are characterized by the BOM in part, or in entirety [1]. While no research was found on predicting the BOM directly from the customer's quotation, research was found in two relevant areas, the optimization of the BOM generation system and predicting the final product cost from the BOM.

High variety production, as seen in jobs shops or mass customization, faces a myriad of management challenges. In order to manage these hurdles, [3] proposes clustering BOMs into generic families. These generic families contain additional operational data giving rise to a bill of material and operation document. A holistic document like this allows for a high degree of flexibility in manufacturing as relationships between component requirements and manufacturing processes are defined for every product. Customization of products are enabled in the generic BOM via parameter values linked to the customer order. The parameters map components, sub-assemblies or sub-sub-assemblies to the generic BOM manifesting a finished, customized product derived from the generic BOM. While this method allows for efficient management of BOM data and a reasonable degree of BOM generation, the BOM variants are predefined. Our proposed method differs from this as BOMs are not predefined but created (trained) from past outputs (BOMs) and past inputs (customer requests). While the algorithm determines rules for prediction, the rules are learnt and not predefined. Therefore, the algorithm may produce variants that have not been considered previously. The output BOM of this paper differs in a second way to [3] as we do not predict any operational data. If this data were readily available it could be included to create a holistic BOM similar to that in [3].

Activity based costing is an accounting methodology used to determine the final cost of a product by allocating the costs of the resources and activities consumed in creating a product. Activity based costing requires analysis of direct costs (e.g., materials consumed) and indirect costs (e.g., maintenance of equipment). For estimators it is often difficult to determine indirect costs as they have not occurred yet or may not explicitly relate to the product. [4] utilizes a neural network to predict the final cost of a product by employing the final product BOM as input data to the network. The model was successful in estimating the indirect costs to manufacture a product and by extension, the final cost to manufacture a product (direct costs are determined from the product BOM hence the algorithm already had this cost). The preceding paper, does not propose a technique for estimating the product BOM and hence the result of this paper would be the input of the model in [4].

Multi-input and multi-output regression is often challenged by volume, velocity, variety and veracity (the four V's framework of big data [5].) Volume expresses the size of the output label space and can initially be large or grow over time posing significant challenges to certain machine learning algorithms. Label spaces can be denoted by binary outputs (multi-labeling classification), ordinal outputs (multi-dimensional classification), or as in the case of this paper, real valued vectors (multi-target regression) [6]. The speed in which new labels are gained in the dataset describes the data's velocity and poses a remarkable hurdle for machine learning. In particular, our application could suffer from high velocity as the bill of materials from the test company includes special components required and supplied by customers. These components are entered into the BOM causing rapid growth in the number of labels. We manage velocity and volume in later sections using a threshold frequency as these customer defined labels are rarely and sporadically repeated. Variety refers to the mixture of data types that can be found in the output labels and veracity can be best described as the quality of the labeling. Our dataset does not suffer from significant challenges in variety as all outputs can be modelled as real valued vectors. The data utilized in this study were the documents created by the BOM estimators, while they may have errors from the estimators themselves there is little noise or missing labels as this would have be cross checked within the company before moving to downstream processes.

Machine learning implementation in the workplace has generally taken three forms, task substitution , task augmentation and task assemblage. Task substitution, as the name suggests, is accomplished via complete automation of the task with minimal human intervention required. Task augmentation is seen when machine learning improves the productivity of the human operator but does not replace the function entirely. Finally, task assemblage occurs when humans and automation are dynamically brought together resulting in new possibilities or abilities. As stated in Section I, we intend to augment the current BOM estimators work flow with machine learning to boost performance. [7] discusses the implementation of machine learning algorithms in businesses and presents a similar implementation to ours. The business case revolves around the automation of trade tables for commodities based on shipping information. Shipping information is gathered from the Automatic Identification System (AIS) as well as contextual data manually transmitted by ships. The ambiguity in some contextual data was a strong motivation for using task augmentation or what the author refers to as "human in the loop". The human operates as a process control parameter by producing ground truth analysis of the data and subsequently correcting the algorithms output . This method worked successfully and after some time the machine learning algorithm was able to produce trade-tables that needed little adjustment from the ground truths. Our implementation will follow a similar approach where the BOM estimators will audit the generated BOMs mitigating any inventory risk and improving the models accuracy.

III. Dataset Description

The data obtained can be represented by a $M \times N + 2$ sparse matrix containing 1347 rows (samples) and 843 columns. Two columns contain text data describing the product. The first of these two columns contains a detailed customer friendly description of the product and the second is a shortened version of the former that is typically used for identifying key attributes about the product during manufacturing. The remaining $N = 841$ columns represent a mix of continuous and discreet variables with one column per component. Job shops are constrained by limited training samples as they are low volume operations and hence the limited sample size of 1347 samples is expected for 3 years of data. Due to the

TABLE I
ILLUSTRATION OF DATASET CONTENTS (FOR A STOOL)

Long Description	Short Description	Legs	Seat	Paint	Back
A red wooden stool with four legs and no backrest	4-legged red stool	4	1	1 gal	0

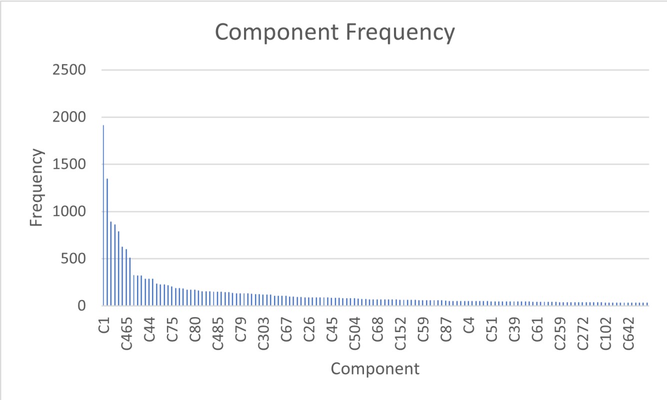

Fig. 1. Occurrence Frequency of different Components

large size of a typical sample we illustrate what information is provided using an illustrative example of a simple stool with contents provided in Table I.

Figure 1 shows the frequency imbalance across the highest occurring components. This behavior is expected as some materials like steel, will be used for most products. Based on company feedback, many of the components listed here were one-time use or customer supplied devices. These special items are not considered part of the company's inventory and would not be commonly procured, consequently, they are not included in our study. Removing these components reduced the label space from 841 to 183 components, we accomplished this by filtering out rare materials with a threshold frequency. For our investigation the business suggested that 15 or less was an acceptable threshold frequency for removing the rare items. These excluded items will be included by the human estimators as they are usually customer specified and are not components existing within the company. The threshold method also acts as a control for the volume and velocity challenges as discussed in Section II. Figure 2 shows the frequency distribution per component, most distributions exhibit a positive skew and the data was standardized for use as outputs of the regression model. Standardization was done per component as significant variations in magnitude can occur.

IV. DESCRIPTION OF MODEL

A. Problem formulation

Let X be a vector describing the product descriptions and as such will contain inputs $[X_1, X_2, ..., X_d]$. Let Y be a vector representing quantities of required components, such that Y contains target variables $[Y_1, Y_2, ..., Y_M]$. (x, y) are samples

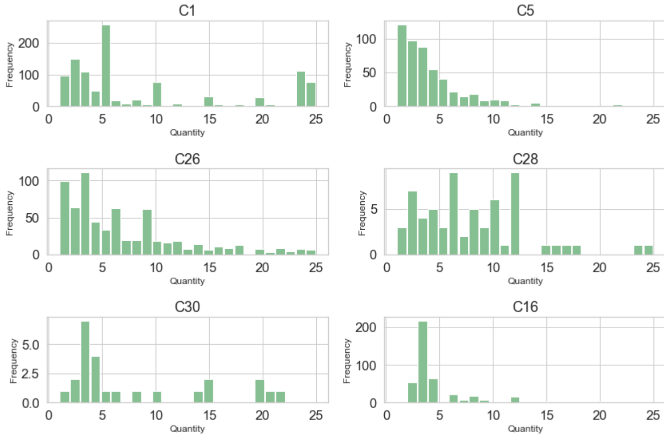

Fig. 2. Frequency distribution of different Components

generated from the domains of X and Y and gives rise to a set of training samples $D = \{(x^1, y^1), (x^2, y^2), ..., (x^m, y^m)\}$, hence our goal is to determine a function, h that maps every sample of x to y, $(h : x \mapsto y)$. From here we enlist the help of neural nets to create these mappings as described in the following section.

B. Model Architecture

Our network architecture uses two input vectors, the first contains the long description. This was reduced using n-grams and uni-grams via TF-IDF (Term Frequency - Inverse Document Frequency) as was done by [8]. If we consider a single sample of a long description, then the Term Frequency of term w in sample d is given by:

$$\mathsf{TF}(w, d) = \frac{\text{Number of times term } w \text{ appears in sample } d}{\text{Total number of terms in the sample } d}$$

The IDF of the term w is given by:

$$\mathsf{IDF}(w) = \ln \left\{ \frac{\text{Total number of documents}}{\text{Number of documents containing } w} \right\}$$

The TF-IDF score of term w in sample d is then given by $\mathsf{TF}(w, d) \times \mathsf{IDF}(w)$.

Using the TF-IDF method caused common words such as "225A" and "120/240V" to be under-represented in the inputs. The second input contained some of these underrepresented words and coupled with a small vocabulary, guided our use of tokenization to reinforce their impact on the neural network. The inputs were independently passed through multiple fully connected layers with relu activation. As stated in the problem formulation, we expected one input vector and as such the inputs are concatenated to one vector and are subjected to multiple fully connected layers with L1 and L2 regularization. Due to the depth of our network, and the associated possibility of unstable gradients we reduce the risk by using the "He-initialization" technique for layer weights as proposed in [9]. Finally, the output layer has a linear activation with the number of neurons equal to the number of components. The number of neurons, network depth, learning rate and regularization

TABLE II
NETWORK HYPER-PARAMATERS

Parameter	Minimum	Maximum
Number of neurons (after concatenation)	1024	4096
Number of layers (after concatenation)	1	10
Dropout layer	0.0	0.2
l1 regularization	0.00005	0.001
l2 regularization	0.00005	0.001
Learning rate	0.000001	0.1

TABLE III
MAE FOR EACH OF THE 5 CV FOLDS

Fold	MAE
01	0.125
02	0.137
03	0.152
04	0.108
05	0.121

parameters are all selected using a non-stochastic bandit formulated algorithm for hyper-parameter selection referred to as Hyperband [10]. The range of parameters given to the network for optimization is shown in Table II. While this increased training time, it ensured a high degree of optimization as percent error translates directly to percent cost for the manufacturer. The Deep Neural Network that was determined through optimization is depicted in Figure 3.

V. ANALYSIS

We use the mean absolute error, MAE, to evaluate the models baseline forecasting performance.

$$\mathsf{MAE} = \frac{1}{N}\sum_{i=1}^{N}\left|\hat{y^i} - y^i\right| \tag{1}$$

A good MAE reassures us that the model is able to predict quantities with an acceptable degree of accuracy. We wanted to be sure of this before including metrics that describe the financial impact of using this algorithm.

The model was trained for 50 epochs using 5-fold cross-validation. The average MAE was 0.125 over all folds. Our output features were standardized for each component, indicating an average MAE of 0.125 presents an acceptable error in this application. Table III shows the results of each fold.

The objective of our algorithm is not only to perform well on predicting quantities but also to reduce the financial risk of incorrect choices. We took cost into account as follows. For each component i we weighted the error by the unit cost of the component c_i. This cost was transformed using a min-max normalization to ensure that the weights did not dominate the loss function. The loss function will be referred to as the weighted mean absolute error. If we denote the predicted and

TABLE IV
% COST ERROR FOR EACH OF THE 5 CV FOLDS

Fold	% Cost Error
01	1.76
02	1.72
03	1.81
04	1.68
05	1.51

actual quantities of component i by $\hat{y}_i$ and y_i respectively then this error for a given test sample is given by

$$\mathsf{WMAE} = \frac{1}{N}\sum_{i=1}^{N}\left|\hat{y^i} - y^i\right|c_i \tag{2}$$

As a result of component weighting, functions such as stochastic gradient decent can become unstable and we opt for the ADAM algorithm to perform gradient decent and increase model stability during training. The weighted loss gives reasonable results when translated into financial savings for the company, we determine this using the percent error in cost defined by equation 3

$$\%\mathsf{CostError} = \frac{\sum_{i=1}^{N}\left|\hat{y^i} - y^i\right|c_i}{\sum_{i=1}^{N}|y^i|c_i} \tag{3}$$

Training and evaluation was implemented with the preceding loss function and table IV shows the results

The average cost error of 1,7% represents an acceptable financial risk to the company and it is worth reiterating that this method will be deployed in augmentation with the existing BOM estimators. The tool is intended to accelerate their estimation process as they will audit the results of the algorithm before releasing the BOM to downstream business processes. Hence, the error found above represents a worst case scenario provided the BOM estimators did not audit any values. Our implementation of this model is available at [11].

The business case discussed in this paper can be sequenced into two problems. The first being a multi-label classification problem where given a corpus, we must select appropriate labels (components) and secondly, a regression problem for predicting the quantity of each label. Previously, we used multi-input multi-output regression to predict the values for each label but this approach lacks interpretability. It is unclear to what degree the neural net was capable of deciding on what components are needed and which are not. The MAE metric is not able to readily answer this, hence, we test a deep neural net optimized for higher recall to understand how well the weights could be trained for multi-label classification for this data. The test model utilized both inputs as the previous model and a sigmoid output layer as shown in Figure 4. Our trial produced an average F_1 score of 0.74, indicating the correct labels could be predicted with some degree of confidence by a deep neural net. Table V shows the F_1 scores for all 5 folds.

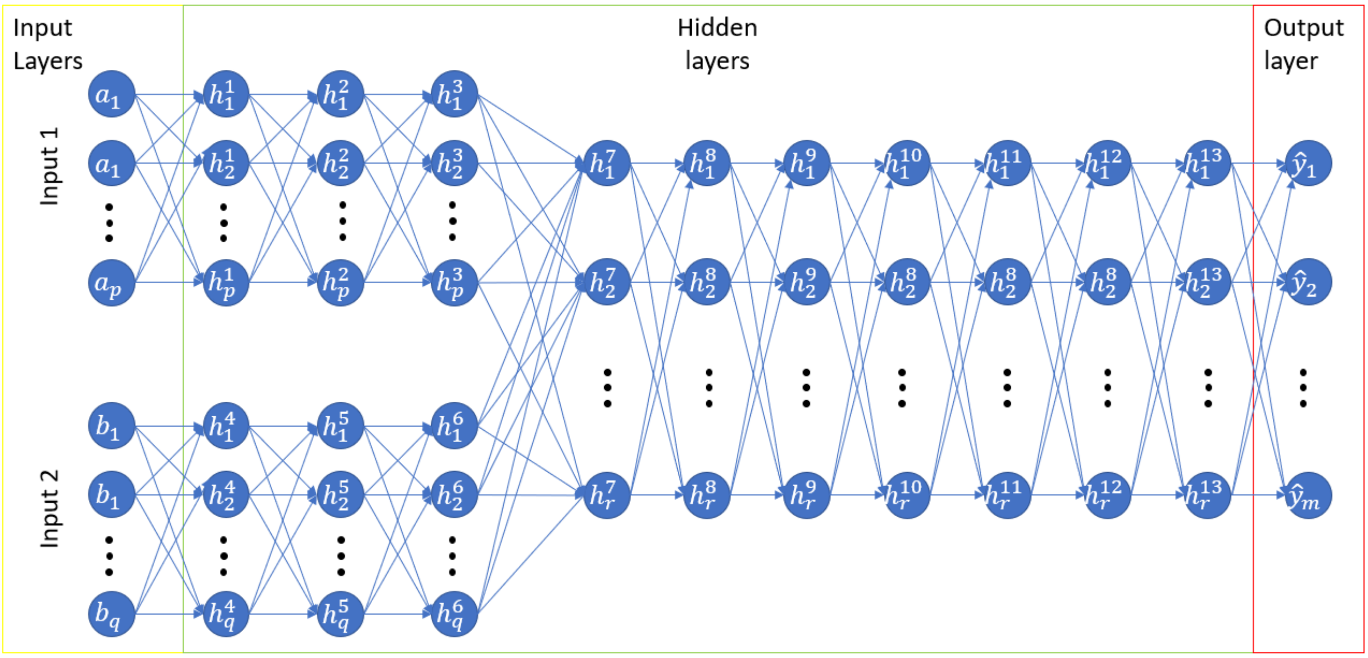

Fig. 3. Model architecture

TABLE V
CLASSIFICATION METRICS FOR 5 CV FOLDS

Fold	Precision	Recall	F_1
01	0.80	0.73	0.76
02	0.78	0.65	0.71
03	0.77	0.70	0.73
04	0.81	0.69	0.75
05	0.79	0.68	0.74

TABLE VI
MAE FOR TWO-PART MODEL

Fold	MAE
01	0.131
02	0.144
03	0.108
04	0.102
05	0.132

At this stage, it was possible to train individual regression models for every component and coupling this with the label classifier we developed an alternative model for solving our business problem. Figure 4 shows the architecture of this model and it is worth noting that the multi-label classifier and regression components were trained independently of each other. We make predictions from the input data using both parts of the model, the first part gives the applicable labels and the second part gives the quantities. Both parts are combined by the Hadamard product to give a final result. Five fold cross validation yielded an average MAE of 0.124, while this provides a precision boost over the multi-input multi-output regression model used previously, it is not economical when compared to the increase in computational requirements. The detailed results of each fold are given in table VI

This deconstructed method was developed to add interpretation to the neural nets ability to solve the problem. While interpretation is important, the comparable MAE and low % cost error from the multi-input multi-output regression make it infeasible as a first choice.

VI. CONCLUSION AND FUTURE WORK

We presented a method for estimating an engineering bill of materials given the customer's quotation. Our model is tested with real world data from a job shop that manufactures electrical panels and we utilize multi-output regression with a custom loss function to minimize financial risk to the company. The loss function is defined by computing the mean absolute error and weighting it by the min-max scaled cost of each component. The financial impact was determined by the percent cost error and is used as an evaluation metric alongside the MAE. Next, we attempted to understand how well a deep neural net can perform on this limited data in selecting appropriate labels. We trained a multi-label neural net resulting in an acceptable F_1 score indicating that the model could adjust it's weights sufficiently to predict the high number of label spaces even with the limited data. We then extended this by creating regressions for each label and developed an alternative two-part model that resulted in a similar performance as the multi-output neural net. This two-part model was computationally expensive but added deeper interpretation to the neural nets ability.

Future work can be done with the methodology presented in this paper by combining the works of [4] with ours, resulting in deeper automation of manufacturing processes. Blending the work in [4] with ours would compute the final cost of a product starting from the customer's quotation. A comparison of methods such as multi-output support vector regression or multi-target regression trees can be used for different BOM structures such as those described in [3]. Class imbalance was observed in the components frequency distribution and while we weighted the loss function based on the product cost, the algorithm can be improved using under-sampling or over-sampling methods. The cost function could be improved by weighting the impact of over-purchasing higher than the impact of under-purchasing

REFERENCES

[1] G. Ghiani, A. Grieco, A. Guerrieri, A. Manni, E. Manni *et al.*, "Large-scale assembly job shop scheduling problems with bill of materials: models and algorithms," *WSEAS Trans. on business and economics*, vol. 12, pp. 161–172, 2015.

[2] J. Heizer, B. Render, and C. Munson, *Operations management: sustainability and supply chain management*. Pearson, 2017.

[3] J. Jiao, M. M. Tseng, Q. Ma, and Y. Zou, "Generic bill-of-materials-and-operations for high-variety production management," *Concurrent Engineering*, vol. 8, no. 4, pp. 297–321, 2000.

[4] Z. Yongqian, C. Meng, Y. Xuebo, and Z. Jin, "A manufacturing cost estimation method based on activity-based costing," in *2010 International Conference on Mechanic Automation and Control Engineering*. IEEE, 2010, pp. 3476–3479.

[5] D. Xu, Y. Shi, I. W. Tsang, Y.-S. Ong, C. Gong, and X. Shen, "Survey on multi-output learning," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 7, pp. 2409–2429, 2020.

[6] H. Borchani, G. Varando, C. Bielza, and P. Larrañaga, "A survey on multi-output regression," *WIREs Data Mining and Knowledge Discovery*, vol. 5, no. 5, pp. 216–233, 2015.

[7] T. Grønsund and M. Aanestad, "Augmenting the algorithm: Emerging human-in-the-loop work configurations," *The Journal of Strategic Information Systems*, vol. 29, no. 2, p. 101614, 2020.

[8] S. S. M. M. Rahman, K. B. M. B. Biplob, M. H. Rahman, K. Sarker, and T. Islam, "An investigation and evaluation of n-gram, tf-idf and ensemble methods in sentiment classification," in *International Conference on Cyber Security and Computer Science*. Springer, 2020, pp. 391–402.

[9] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[10] D. Xue, "Research on material substitution algorithm during the complex equipment assembling process," in *2010 International Conference On Computer Design and Applications*, vol. 2. IEEE, 2010, pp. V2–259.

[11] B. Kevin, "Dnns for boms," https://github.com/Kelvos/DNNs-for-BOMs, 2021.

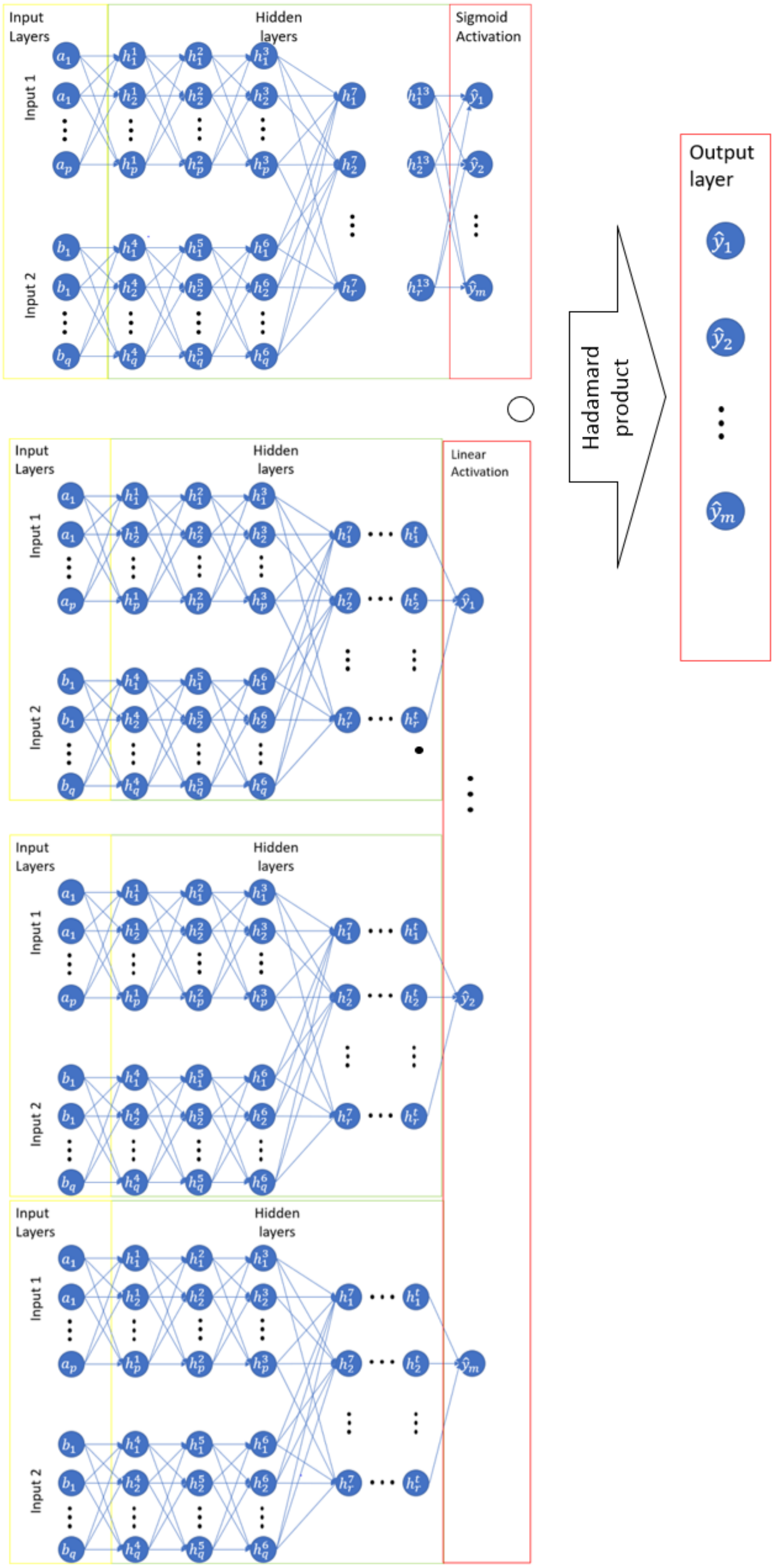

Fig. 4. Model architecture