

## Research Article

# Enhancing Health Risk Prediction with Deep Learning on Big Data and Revised Fusion Node Paradigm

**Hongye Zhong and Jitian Xiao**

*School of Science, Edith Cowan University, Joondalup, WA, Australia*

Correspondence should be addressed to Hongye Zhong; hzhong@our.ecu.edu.au

Received 5 January 2017; Revised 14 April 2017; Accepted 9 May 2017; Published 28 June 2017

Academic Editor: Michele Risi

Copyright © 2017 Hongye Zhong and Jitian Xiao. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

With recent advances in health systems, the amount of health data is expanding rapidly in various formats. This data originates from many new sources including digital records, mobile devices, and wearable health devices. Big health data offers more opportunities for health data analysis and enhancement of health services via innovative approaches. The objective of this research is to develop a framework to enhance health prediction with the revised fusion node and deep learning paradigms. Fusion node is an information fusion model for constructing prediction systems. Deep learning involves the complex application of machine-learning algorithms, such as Bayesian fusions and neural network, for data extraction and logical inference. Deep learning, combined with information fusion paradigms, can be utilized to provide more comprehensive and reliable predictions from big health data. Based on the proposed framework, an experimental system is developed as an illustration for the framework implementation.

## 1. Introduction

The tendency of digital healthcare solutions is to transform the entire healthcare process in more flexible and efficient patterns. The most common applications of digital healthcare include electronic health records, mobile devices, and wearable health devices.

Electronic health records are initially sourced from health checks and diagnostic data of patients. Digitalization facilitates the sharing of the health records across different medical organizations. With these digitalized records, doctors can have a better understanding of the medical history of their patients [1]. However, when information accumulates over time, health records proliferate in large volumes. This causes difficulties for processing, storage, and retrieval. Estimates suggest that health records data may reach 12 ZBs by 2020 [2].

With the prevalence of smart phones, medical apps are available for many useful functions such as electronic prescribing, assessment, clinical decision support, treatment practice management, and self-care. Wearable health devices represent another rapid growth area which is transforming traditional healthcare to active and continuous health management. Research estimates that the number of wearable

health devices could reach 169.5 million globally by 2017 [3]. The physiological sensors embedded in the wearable devices allow for capturing of additional health data. For example, heart rate and blood pressure can also be detected by smart phones.

With the growth of big data, scientists have found increasing value in deep learning and information fusion for the analysis of large volume of dynamic data. Deep learning applies a set of machine-learning algorithms at multiple levels to exploit the different layers of nonlinear information [4]. Information fusion applies continuous processes that collate relevant information to achieve situation awareness, which can support decision-making [5].

With the advent of global healthcare challenges such as age-related health problems and chronic diseases, researchers are actively seeking innovative solutions for disease prevention and health diagnoses in more efficient and economical ways. Using information fusion techniques with the digital health data to produce a viable solution has become a key topic of the industry. The research aims to provide an enhanced framework for developing health risk prediction systems by combining the techniques of deep learning and information fusion.

The rest of the paper identifies the research potential of deep learning and big health data (Section 2), reviews the preliminary methodologies related to the research (Section 3), and introduces an innovative framework to overcome some shortfalls in the peer research (Section 4). Finally, issues with experiment design and performance evaluation are discussed (Section 5).

## 2. Research Potential

*2.1. Deep Learning on Big Health Data.* Scientists have started to realize the insights of the health data held by various repositories and to make decisions for healthcare. However, most of the development potential is still in its infancy. The overall theories and techniques for predictively modeling and analyzing health data have not been adequately developed yet [6]. The conventional data mining methods suffer from accuracy and efficiency issues due to the constraints of data size and data quality [7]. Many conventional methods cannot directly fit in the big data context. In some environments, the concepts are so complex that they have to be learnt by sophisticated procedures. In particular, when there are underlying functional dependencies within a complex nature, we are unable to analytically express the data model in a simple way [8]. The industry is calling for advanced methodologies that allow information extraction from unstructured health data in large volumes [9].

Conventional data mining or machine-learning methods depend heavily on the representation of the data. In complex environments, it is difficult to extract high-level and abstract features for data representation due to variant factors [10]. Deep learning is introduced to solve this problem whereby it represents a complex concept by combining several simpler concepts. Deep learning is a machine-learning paradigm with a set of algorithms that attempt to learn a complex matter by mapping it to different levels of abstraction [4]. It has also been proposed for medical knowledge modeling and assisting in decision-making on healthcare issues. It is valuable for constructing health diagnosis models in a cost-efficient manner. In contrast, the cost of building an expert health system in the conventional approach was usually unaffordable [11].

Various computing models are being rapidly developed and applied in deep learning. Neural network is one of the most common models that can be used for implementing deep learning systems. They are highly adaptable to feature extraction from various situations [12].

Although deep learning can extract features from complex data, in a big data context, data combination must be performed before analytics take place. As information fusion can provide various methodologies for merging data from multiple sources, embracing the concept of information fusion within deep learning domain can enhance big data analytics with more systematic designs and more efficient processes.

*2.2. Enhancement for AIA Vitality.* AIA is an insurance company that owns a large customer base over Asian-Pacific countries including Australia and China. AIA recently

introduced a science-backed wellness program called AIA Vitality that attempts to motivate the customers to maintain their health status with rewards and benefits provided by the partner organizations (see Figure 1).

The members of AIA Vitality can understand their current health status by uploading their health figures from their mobile and health devices, performing health assessments at the partner medical centers, or receiving periodic health reports generated from the collected formation. The members can improve their health by participating in health activities and meeting the health goals proposed by the system. Health reports will be generated periodically for members by analyzing the health records collected from different channels and the health activities undertaken by members. Based on their health status, members can be rewarded by the partner organizations.

The health risk predictions contained in the health reports mainly come from the expert opinions of the partner medical centers and in-database analysis rules. Health experts can log in to the system, review the health records of the members, and input their opinions. However, this process has been criticized for the overconsumption of both money and time. The health prediction rules set in the database are triggered when the health figures of a member meet certain conditions. The triggered rule will lead to a health risk being added in the report. For example, when the blood glucose of a member at a certain age reaches a certain level for a certain duration, the risk of diabetes will be identified in the member's health report. However, this process can only provide limited prediction categories with insufficient accuracy. We propose overcoming these issues and enhancing the health prediction by applying deep learning and information fusion on the big data in the system. Applying deep learning to big health data, the hierarchy of features can be learnt by establishing high-level features in terms of health prediction from low-level health datasets [13].

## 3. Preliminary Methodologies

To date, the healthcare industry has realized the potential benefits that can be gained from big data analytics. Recently, it has attracted extensive research on the architectural design and implementation strategies of big data analytics in the area of health data.

*3.1. Architectural Design.* The fundamental requirement of big data analytic systems is to deal with the volume, variety, and velocity of data coming from sources sharing the same context. Heterogeneous data is generated and stored in multiple sources such as relational databases, XML (Extensible Markup Language) documents, and RDF (Resource Description Framework) [3]. Transforming these heterogeneous datasets into understandable and sharable knowledge requires services that collect, prepare, and process these datasets.

The architectural design for a big data analytic system in healthcare is similar to that of a traditional big data analytics project. NIST (National Institute of Standards and Technology) provides a reference on architectural design for

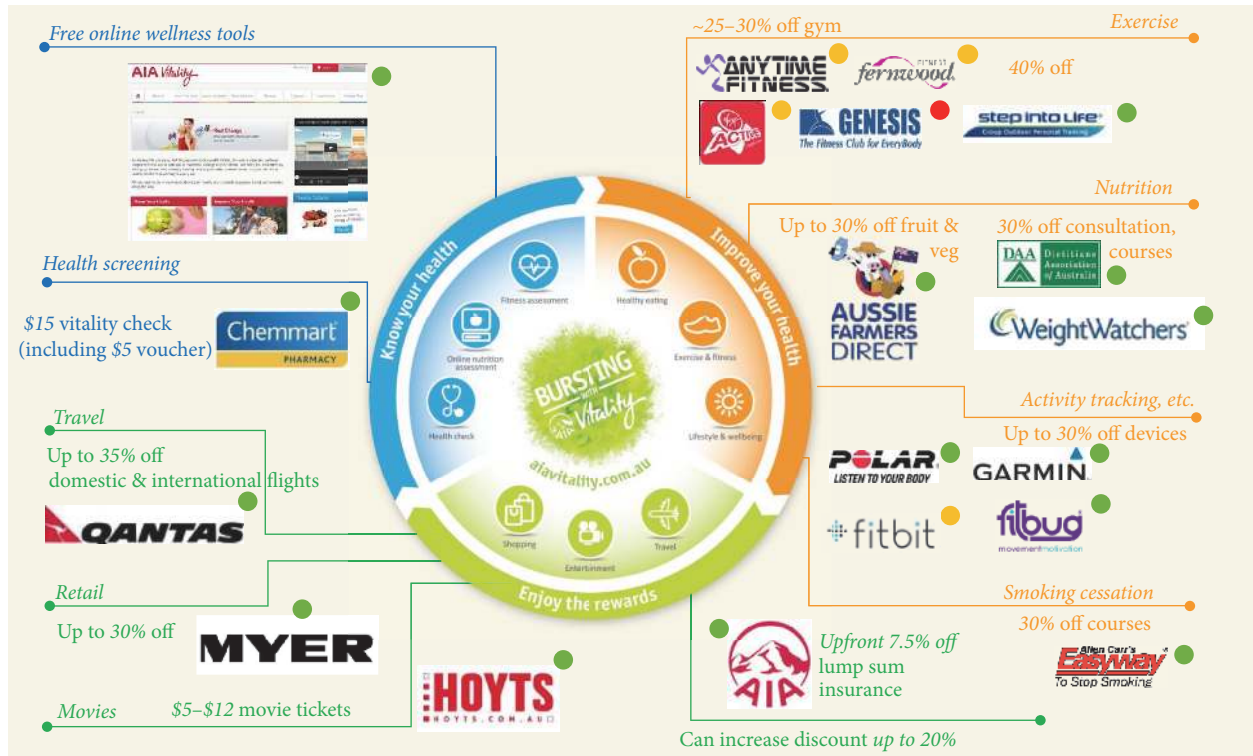


FIGURE 1: AIA Vitality program.

big data systems [14]. There have been several attempts to adapt the NIST reference to healthcare domains. Among those, one worth mentioning is the architectural design for big data healthcare analytics by Wang et al. (see Figure 2) [15]. This design functionally divides a big data analytic system into five layers, namely, (1) Data Layer, (2) Data Aggregation Layer, (3) Analytics Layer, (4) Information Exploration Layer, and (5) Big Data Governance Layer.

Data Layer contains all the data collected from various sources that can provide the insights to support daily operations and solve business problems. The data from various sources can be structured (e.g., database records) or unstructured (e.g., log files or images).

Data Aggregation Layer is responsible for data acquisition from the various data sources and transformation of the data into a specific standard format that is suitable for data analysis. Data transformation is often a major obstacle for implementing big data, as the data characteristics can vary dramatically.

Analytics Layer is responsible for performing appropriate analytics on the aggregated data based on the fundamental analysis goals. Recent advances in cloud computing and big data analysis, respectively, provide distributed data storage and computing techniques, including MapReduce, stream programming, and in-database analytics, which allow for the handling of vast amounts of data in a more reliable and efficient manner [3].

Information Exploration Layer generates outputs to make the analytic results comprehensible for users. The outputs can

be visualization reports, monitoring of information, or health predictions to help users make decisions.

Big Data Governance Layer is a component that manages all the other logical layers. Its functions include data accessibility, data lifecycle, and data security managements. These functions ensure the availability of the data for processing and protect the privacy of the information owners.

**3.2. Data Mining Methods.** Data mining is the process of information extraction and pattern discovery from large amounts of data. This concept is expanded by machine learning to transform data into intelligent action [16]. Data mining is being used in the emerging healthcare area to manipulate clinical and diagnostic data and thereby provide reliable disease detection and healthcare systems [7]. The data mining methods used in the health industry fall predominantly into two categories: cluster analysis and classification. They provide mathematical foundations for the processes of learning and prediction.

Cluster analysis is an unsupervised machine-learning method that can organize a group of objects with similar characteristics into distinct categories. The main aim of clustering is to find structure within a given set of data [17]. Clustering is useful in healthcare for identifying the association between risk factors and health. For example, the research of Vermeulen-Smit et al. applied clustering patterns to explore the relationship between health risk behaviors and mental disorders [18]. Clustering is also useful for anomaly detection, revealing the most anomalous or unusual data from the cohort dataset.

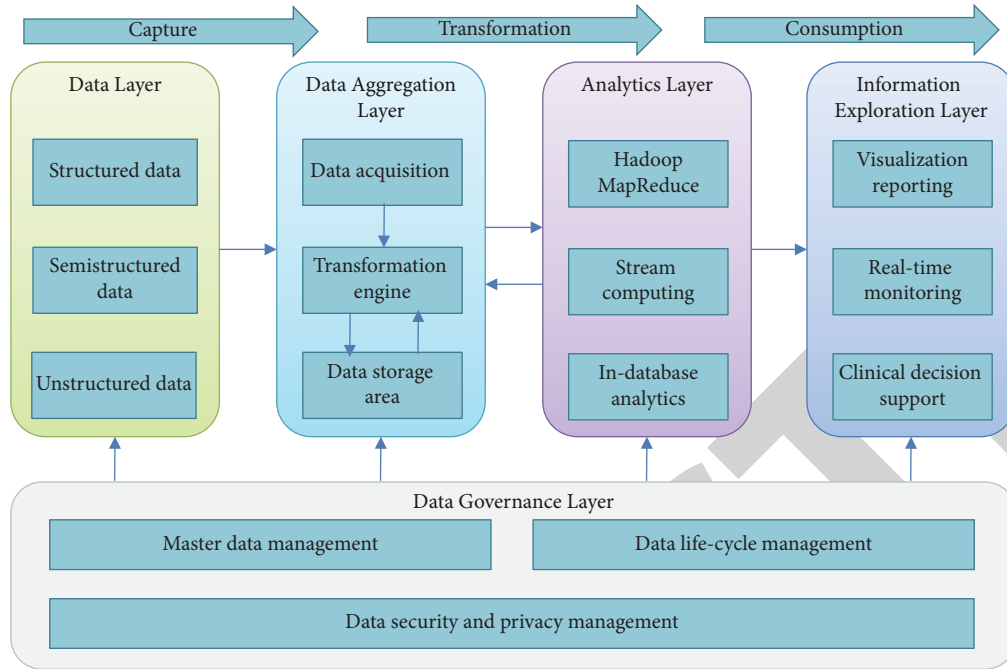


FIGURE 2: Big data architecture for healthcare.

Classification is a supervised learning process for prediction of the class for a given unlabeled item in a finite set of predefined classes based on a training set of data containing observations [19]. The most common classification algorithms include Decision Tree,  $K$ -Nearest Neighbor (KNN), Logistic Regression (LR), Support Vector Machine, and Bayesian Classifier. Classification is useful for predictive learning in health fields. For example, Bayesian Classifier was introduced by Kazmierska and Malicki to the diagnosis of brain tumor patients and help in the clinical treatment of the patients [20].

Decision Trees can classify data items into a finite number of predefined classes.  $K$ -Nearest Neighbor classification assigns labels of predefined classes to the test items by finding a group of  $K$  items in the training set in the neighborhood. Logistic Regression computes a linear combination for the input item by the logistic function with a given set of features. Support Vector Machine separates the test items by finding the separating lines with maximum distance to the points of the different groups of items [21].

Bayesian Classifier is a supervised and probabilistic learning method. The training data is used to calculate an observed probability of each outcome based on the evidence provided by feature values. The observed probabilities are then applied to predict the most likely class for the new features [16].

**3.3. Bayesian Fusion and Neural Network.** By consolidating between different information sources, information fusion can perform data mining to reduce uncertainty and achieve a better understanding of the information. By adopting Bayesian statistics, it can provide several methods for information fusion [22], for instance, the Bayesian Classifier above.

Bayesian fusion algorithms use the knowledge of Bayesian Inference to make inferences about the identity of events in the observation space [23]. Bayesian Inference is the process that applies a probability model to a set of data and evaluates the resulting probability distribution for the fit of the model towards the observed and unobserved data [24]. The Bayesian Inference process can be applied to the fusion from multiple information sources.

In the inference process, a.k.a. Bayesian fusion process (Figure 3), each of the data sources provides a hypothesis on the observation and the source-specific algorithm. The hypothesis  $H^n (k = 1, 2, \dots, n)$  is used to estimate the probability of the type of each entity by the likelihood function  $P(H^k | O_i)$ , where  $i$  denotes the fact type and  $O_i$  the entity from a data source. The probabilities of the data sources will be combined by using the Bayesian Inference Function. The output of the inference is the combined probability  $P(H^1, H^2, \dots, H^n | O_i)$ . Decision Logic is used to optimize the combined probability when there are constraints (e.g., thresholds) on the final output. The final output of the process is called the Fused Identity of  $O_i$ .

The ground of the Bayesian Inference is Bayes' Rule, which is a probability-based reasoning discipline. Bayes' Rule can be derived by evaluating the probability of hypothesis  $H$  conditioned on knowing event  $E$ , which is defined as

$$P(H | E) = \frac{P(H, E)}{P(E)}, \quad (1)$$

where  $P(H, E)$  denotes the probability of the intersection of hypothesis  $H$  and event  $E$  [23].



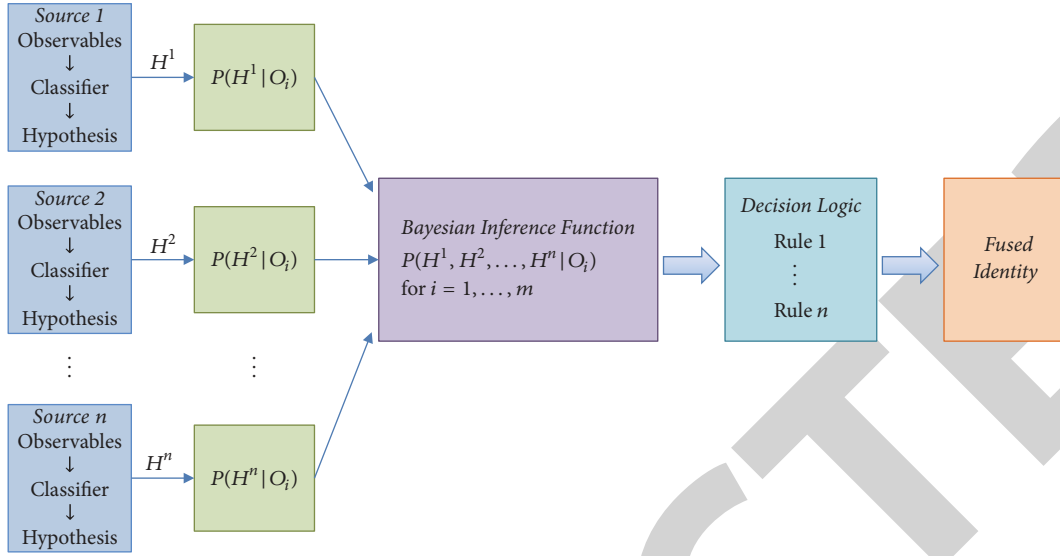


FIGURE 3: Bayesian fusion process.

The inference applied in Bayesian Classifier can be defined as

$$\text{Assign } x \text{ to } C_i = \arg \max_{C_i} P(C_i | x), \quad (2)$$

where  $C_i$  ( $i = 1, 2, \dots, m$ ) denotes a given set of  $m$  classes,  $P(C_i | x)$  denotes the respective posterior probabilities, and  $x$  represents an unknown feature vector. The unknown attribute  $x$  is identified by assigning it to the class for which the posterior probability becomes maximum [8].

Bayes' Rule enables the correct estimation of a prior probability for an unknown event when the probability of the evidence about the event can be found. This works on the assumption that the unknown event and the evidence are independent. However, in the real world there are many unknown events and different types of evidence, some of which are related to each other [25]. Therefore, the Bayesian Network is introduced to represent the joint probability model among given variables by a directed acyclic graph, where the nodes of the graph represent random variables and the directed edges represent direct dependencies between the variables [26]. Mathematically, this type of structure is called a directed acyclic graph (DAG). The joint probability of the model can be calculated by

$$\begin{aligned} P(U) &= P(X_1, \dots, X_n) = \prod_{X_i \in U} P(X_i | X_1, \dots, X_n) \\ &= \prod_{i=1}^n P(X_i | \text{Pa}(X_i)), \end{aligned} \quad (3)$$

where  $U = \{X_1, \dots, X_n\}$  is the set of random variables on interest called universe and  $\text{Pa}(X_i)$  denotes the parental variables of variable  $X_i$ . Referring to a DAG, it represents the  $i$ th node in the graph corresponding to the factor  $P(X_i | \text{Pa}(X_i))$  [27]. Bayesian Network is a fusion approach that is suitable for the uncertainty measurement for the extracted properties with graphical structure and probability calculus [28].

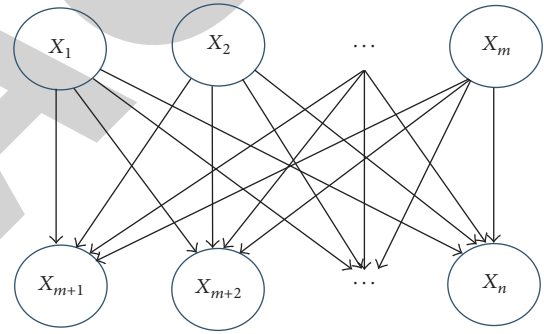


FIGURE 4: Example of Bayesian Network.

A Bayesian Network as shown in Figure 4 represents the association  $\{X_1, X_2, \dots, X_m\} \Rightarrow \{X_{m+1}, X_{m+2}, \dots, X_n\}$ , where the nodes  $X_i$  represent the variables and the edges represent the direct influence. The joint probability distribution of the network can be defined as

$$\begin{aligned} P(X_1, X_2, \dots, X_m, X_{m+1}, \dots, X_n) &= \prod_{i=1}^n P(X_i | \text{parents}(X_i)) \\ &= \prod_{i=1}^n P(X_i) \prod_{j=m+1}^n P(X_j | X_1, X_2, \dots, X_m) \\ &= \prod_{i=1}^n S(\{X_i\}) \prod_{j=m+1}^n \frac{P(\{X_j, X_1, X_2, \dots, X_m\})}{P(\{X_1, X_2, \dots, X_m\})} \\ &= \prod_{i=1}^n S(\{X_i\}) \prod_{j=m+1}^n \frac{S(\{X_j, X_1, X_2, \dots, X_m\})}{S(\{X_1, X_2, \dots, X_m\})}, \end{aligned} \quad (4)$$

where  $S$  is the support of an itemset.

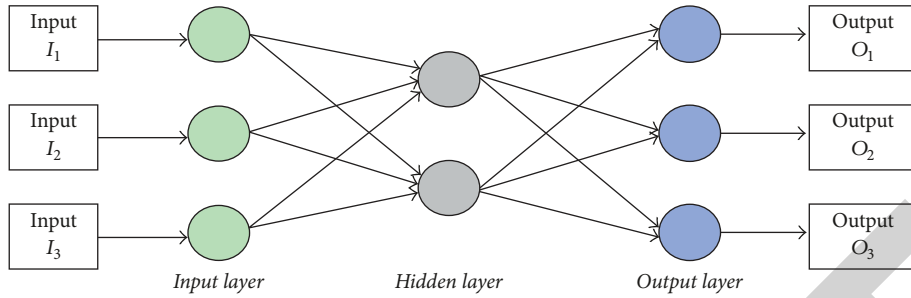


FIGURE 5: Three-layer feedforward network.

In information fusion, the selection of features can be very complex under certain circumstances. For example, a speech recognition task in a noisy environment is difficult due to the interference of irrelevant features [29]. On such occasions, feature extraction approaches can be used to extract the main features or characteristics of the situation [12]. Neural network can provide a flexible method to capture essential features from data containing noise and partial information [30].

Neural network is a function-approximation technique that with a set of given inputs and a desired set of outputs can be trained to map observed data to a function [31]. A standard structure for a neural network is a multilayer feedforward network [12].

A simple example is a three-layer perceptron network (Figure 5), which consists of the input layer, the hidden layer, and the output layer. The nodes in the input layer supply the input signals to the nodes in the hidden layer. The connections between the nodes in the input layer and the hidden layer represent the weights between the two layers, which need to be determined by training. The output signals of the hidden layer are used as inputs to the third layer. The outputs of the nodes in the output layer constitute the network output. The network output is usually an output prediction, a function approximation, or a set of classification outputs [30].

Each node in the hidden layer simulates a brain neuron, which is composed of a summation function and an activation function (see Figure 6). The summation function computes a linear combination of the weighted input signals [27]. The function can be defined as

$$u_j = \sum_j (\cdot) = \sum_{i=1}^n w_{ij}v_i + b_j, \quad (5)$$

where  $v_i$  denotes the  $i$ th input signal in the total  $n$  input signals,  $w_{ij}$  denotes the weight of the input signals associated with the  $j$ th hidden node, and  $b_j$  is the externally applied bias.

The output of the summation function is passed to the activation function that squashes the permissible range of the output signal to a finite value. The activation function is introduced to provide nonlinearity into the network, which is usually differentiable and chosen depending upon the requirements of the case [32].

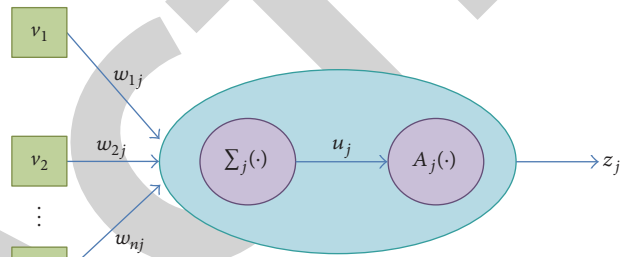


FIGURE 6: Neural node structure.

#### 4. Deep Learning Fusion Framework

Deep learning is not an entirely new technology. Examples of the application of deep learning in healthcare can be found in several publications, for instance, on mammogram analysis [33] and video pornography detection [34]. However, these applications are mostly limited to image analysis. Additionally, these studies have not been widely adopted, as the architecture of deep learning is more complex than conventional data mining systems, and the previous studies failed to propose system designs that are compatible with the analytics in a production environment.

Deep learning systems require more computing power and memory storage. Recently, with the broad uptake of cloud computing, deep learning has been given new impetus by offloading the computation onto the cloud [35]. This allows an analytic system with complex learning algorithms to be hosted on the cloud in a cost-efficient manner.

A multilayered architecture is used in deep learning to map the relations between inputs features and the outcomes. This architecture makes deep learning more suitable for analyzing a large number of variables. The model containing many levels of nonlinearities excels at identifying complex patterns in data [32]. However, in a big data environment, data that comes from multiple sources and from multiple structures need to be fitted into multiple models. The fusion of the models should be performed at a specific level of the architecture to achieve the overall learning purpose.

The most fundamental fusion model is called situation awareness (SAW). SAW can be divided into three layers, in terms of (1) perception of elements in current situation,

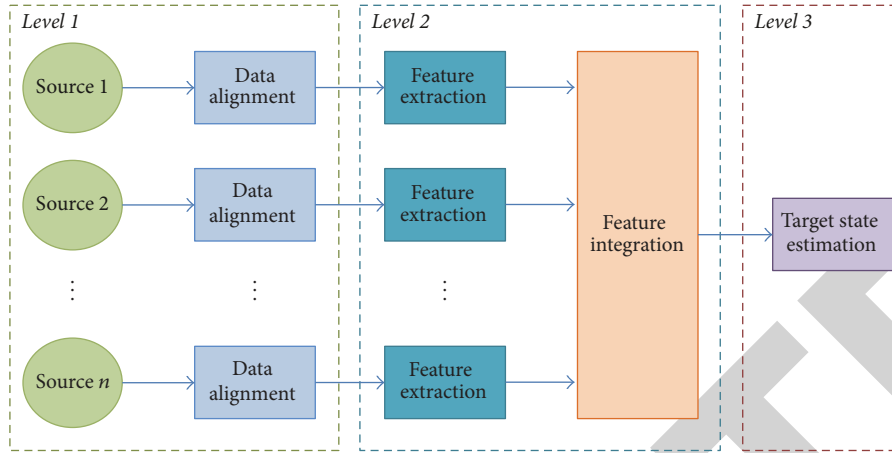


FIGURE 7: Deep Health Analytic Framework.

(2) comprehension of current situation, and (3) projection of future states [36]. The research of Sundareshan and Wong [37] extended the concept to the extraction of features from multiple data sources. The research of Mitchell [38] developed a parallel network for the fusion of multiple data inputs. These research projects provide useful guide on structuring data analysis system. However, they lack details on implementing the data process components. Steinberg et al. introduced the fusion node paradigm that indicates the main processes contained in a fusion system in terms of fusion, association, and alignment [39]. This paradigm has proven to be useful for developing automatic fusion systems that involve selecting the data flow among the fusion nodes (i.e., how to batch data for association and fusion processing). We integrate deep learning with the fusion processes to perceive the current situation and predict the future status of an individual’s health from the big data.

Integrating fusion paradigms with preliminary big data techniques, we propose a framework suitable for deep learning on big health data, namely, the Deep Health Analytic Framework (DHAF, Figure 7). The framework is composed of three levels, which correspond functionally to the 3-layer structure of the SAW model. At the first level, the fusion processes aim to acquire the status, attributes, and dynamics of the relevant information elements in the environment. The processes consist of the data acquisition from multiple data sources and the alignment of the raw data. The eHealth data usually consists of various file objects, including images and PDF files. Although there are several commercial ETL (Extract, Transform, and Load) software programs, for example, Informatica, which allows the integration of data from multiple sources, the existing ETL systems are usually very costly and provide limited support of data types. The data alignment using machine-learning approaches is introduced to provide a more flexible solution. The proposed alignment processes involve classifying the information elements and restructuring them to representations compatible for further analysis [12]. The data alignment for each source is achieved by a fusion node that encompasses several components as described in Figure 8.

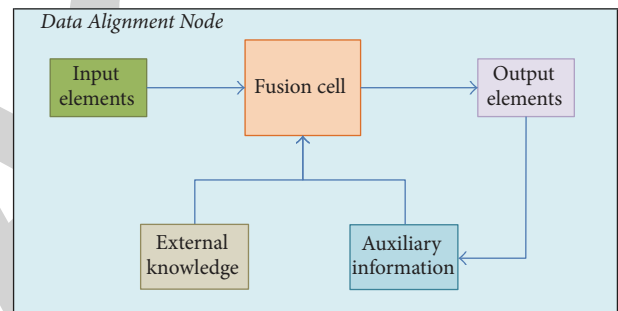


FIGURE 8: The fusion node of data alignment.

The input elements are the original information objects from a data source. Fusion cell is the smallest granular component of information fusion systems. It acts as an intermediate conceptual level between the different individual information elements from the data source and the global information processing systems [40]. Fusion cell refines the input data elements with the constraints or indicators set in the external knowledge and auxiliary information and ensures the quality of the output elements meet the requirements for further fusion processing. The external knowledge contains operation rules that can constrain and guide the fusion process. These rules contain the information required to process data in specific formats. For example, if the input data is in XML format, fusion cell will use the XML specification information for the data analysis. The auxiliary information stores additional contextual information to support the fusion process. The results derived from the fusion processes can also be recorded in auxiliary information to enhance future fusion.

The process method of fusion cell is usually a classifier, and the outputs are maps of classes and of confidence levels [41]. We use neural network to construct the fusion cell. According to the research of Amirani et al., we can compute Byte Frequency Distribution (BFD) and Principal Component Analysis (PCA) to identify different data types. BFD is the occurrence pattern of each byte value within

a file content. It can be used to discover the files derived from different data sources by detecting the file type fingerprints. PCA is a multivariate analysis technique that describes data with an orthogonal transformation of the coordinates. It can represent an approximation of a data structure with a subset of its primary components by dimensionality reduction [42]. A  $d$ -dimensional dataset  $M = \{M_n \in R^d \mid n = 1, \dots, N\}$  can be approximately represented by a smaller  $k$ -dimensional dataset  $Z = \{Z_n \in R^k \mid n = 1, \dots, k\}$ .  $Z$  can be obtained by  $ME_k$ , which means projecting  $M$  onto the eigenvectors corresponding to the largest eigenvalues.  $E_k$  is the first  $k$  column of the eigenvectors  $E$ .

An autoassociative neural network is used for classifying the data type. A feedforward network is trained to produce an approximation of the identity mapping between inputs and outputs using back-propagation algorithm. The network has  $d$  input and  $d$  output nodes and a single hidden layer with  $k$  nodes and linear activations. In the training phase, the PCA projection matrix is calculated after the BFD of the input dataset is normalized. The normalization is to scale the data into the interval of  $[-1, 1]$ . The output dataset from PCA is fed to the neural network. As it is a nonlinear process, the hyperbolic tangent is chosen as the activation function. The function can be defined as

$$z = A(u) = 1 - \frac{2}{1 + \exp(2u)} \in (-1, 1), \quad (6)$$

where  $u$  denotes the input dataset and  $z$  the output dataset. The training outputs are compared with the desired outputs of the data types, and the synapse weights are adjusted accordingly to reduce the errors. After the training, the weight of the synapses is saved in the auxiliary information of the fusion node. Once the data type has been determined, the format specification in external knowledge will be used to verify the identification results, extract the metadata of the file object, and normalize the content of the object to a common representational format. If the identification is incorrect, an error will be recorded in auxiliary information to enhance future identification. For instance, an object that has been identified as a PNG image will be transformed to a standard size and color, and its content data and metadata will be stored in XML format. The metadata should contain information related to the environment such as the transaction time and spatial references. The transformation of the source data facilitates the analysis of information elements in a consistent set of units and coordinates for further processing [12]. The aligned data will be ready for the second-level fusion.

At the second level of the framework, fusion processes are intended to interpret and integrate the information elements under the current situation. The processes in this level include feature extraction and feature integration. The feature extraction is to acquire perceptual features containing the maximal information concerning the prediction objectives in the next fusion level. The extracted information is organized on the basis of the time sequence. Through the extraction process, the data quantity and data dimension are reduced to increase the efficiency and accuracy of further analysis. A fusion node for the feature extraction encompasses several components, as shown in Figure 9.

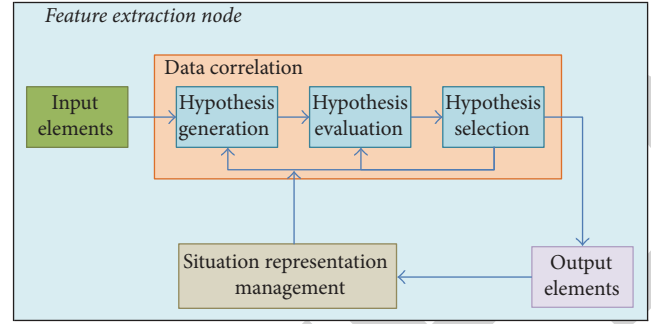


FIGURE 9: The fusion node of feature extraction.

The main process in the extraction node is Data Correlation. This determines which input elements associate with which elements currently in the situation representation being maintained by the node. The input elements for the association are the aligned data in the common format and the time coordinates that allow the elements to be tracked along with time. The correlation process is accomplished through three functions, namely, Hypothesis Generation, Hypothesis Evaluation, and Hypothesis Selection. Through the correlation process, the input data becomes normalized and fits the situation representation.

Hypothesis Generation identifies feasible association candidates that can be derived from the input elements [39]. In this function, the hypothesis of the associations between the input elements and the health risks will be generated. For example, running records taken from a wearable device can be used as the input elements, and a hypothesis can be generated to represent an association between the pattern of steps run and a health risk (e.g., obesity). Hypothesis Evaluation computes the confidence of the identified candidates based on the defined metrics to determine the associations. For example, the confidence of association between the running record and the obesity will be calculated.

The confidence calculation can be achieved by Bayesian Association Rule Mining algorithm [43]. For an equivalent timeframe, let  $A$  be an itemset (e.g., a health device record) consisted of items  $I_1, I_2, \dots, I_m$ , and let  $R$  be a health risk consisted of items  $I_{m+1}, I_{m+2}, \dots, I_n$  (e.g., the historical diagnoses related to the health risk). The association  $A \Rightarrow R$  can be represented in a Bayesian Network, where  $I_1, I_2, \dots, I_m$  are the Boolean variables in correspondence with  $X_1, X_2, \dots, X_m$  (see Figure 4) and  $I_{m+1}, I_{m+2}, \dots, I_n$  correspond to  $X_{m+1}, X_{m+2}, \dots, X_n$ .

The confidence of association  $A \Rightarrow R$  is defined as  $P(R | A)$ . The confidence can be computed with Bayesian Network, namely, Bayesian Confidence (BC), which can be represented as

$$\begin{aligned} \text{BC}(A \Rightarrow R) &= P(R | A) = \frac{P(A, R)}{P(A)} \\ &= \frac{P(X_1, X_2, \dots, X_m, X_{m+1}, \dots, X_n)}{P(\{X_1, X_2, \dots, X_m\})} \\ &= \frac{P(X_1, X_2, \dots, X_m, X_{m+1}, \dots, X_n)}{S(\{X_1, X_2, \dots, X_m\})}. \end{aligned} \quad (7)$$



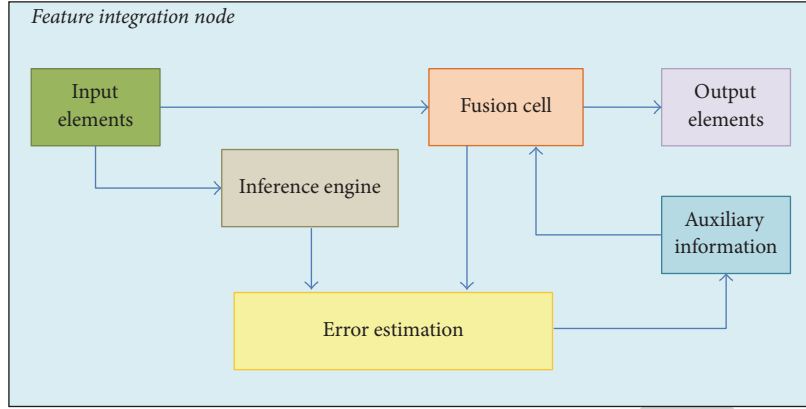


FIGURE 10: The fusion node of feature integration.

Hypothesis Selection determines which output associations the fusion system should retain and use for state estimation. For example, if the association of a hypothesis exists, that is, the calculated confidence value is greater than 1.0, the association should be updated into the situation model. The maintenance of the situation model is required to compute the probabilities of all the events, that is, the probabilities for the occurrence of various health crises. Let us denote the probability of a health risk as  $P(R)$  and the conditions on the occurrence of a set of associations as  $\{D_i \mid i = 1, \dots, n\}$ . If the itemset  $A$  introduced above is a selected association, we have  $\{A \in D_i \mid i = 1, \dots, n\}$ . The probabilities of the events can be approximated to an aggregated probability  $P(R \mid D_1, \dots, D_n)$ . The aggregation of the associations requires the computation of a probabilistic model of the joint distribution of  $(R, D_1, \dots, D_n)$ , which can be achieved by building an approximation of the true conditional probability by the use of an aggregation operator. Based on Nu model [44], the approximation can be defined as

$$\begin{aligned} P(R \mid D_1, \dots, D_n) &= \frac{P(R, D_1, \dots, D_n)}{P(D_1, \dots, D_n)} \\ &= \frac{P(R)P(D_1 \mid R)P(D_2 \mid R, D_1) \cdots P(D_n \mid R, \dots, D_{n-1})}{P(D_1, \dots, D_n)} \quad (8) \\ &= P(R) \frac{\prod_{i=1}^n P(D_i, R, \bar{D}_{i-1})}{P(D_1, \dots, D_n)}, \end{aligned}$$

where  $\bar{D}_{i-1} = \{D_1, \dots, D_{i-1}\}$  is the set of all data up to the  $(i - 1)$ th association. This model is useful for combining probability distributions conditioned to each individual association into a joint conditional distribution that can represent all original information. Besides Nu model, there are several alternative approximation methods, such as Tau model [44]. With this aggregation operation, we can compute the approximated probabilities for all the events in different time frames, which can represent the probabilities for various health risks in the eHealth system.

Each feature extraction node manages a fused situation representation for the data from its corresponding data

source. The local fused situation representations for different data sources are to be integrated into a global situation representation, so that the developed model can be used for estimation and decision-making in the next fusion level. This process, namely, feature integration, is achieved by a deep learning method as described in Figure 10.

The input elements for the feature integration embrace the source data normalized by the nodes of data alignment and the probabilities of health incidents fused by the feature extraction nodes. Within a time frame where there are both inputs of normalized data and probabilities, the input probabilities from different sources will be combined to aggregated probabilities by the inference engine, and the input normalized data will be fed into the fusion cell for training purposes. Inference engine is a component that updates probabilities using inference scheme and evidence. Each piece of evidence updates the probability of a set of hypotheses calculated via Bayesian rule. The calculation can be defined as

$$P(A \mid B) = \frac{P(B \mid A)P(A)}{P(B)}, \quad (9)$$

where  $A$  and  $B$  are events that are not necessarily mutually exclusive,  $P(A \mid B)$  is the conditional probability of event  $A$  occurring given that event  $B$  has occurred [36]. If there are  $m$  mutually exclusive and exhaustive hypotheses  $H_1, \dots, H_m$  and  $n$  possible occurring events  $X_1, \dots, X_n$ , the probability can be updated as

$$P(H_i \mid X_j) = \frac{P(X_j \mid H_i)P(H_i)}{\sum_{k=1}^m P(X_j \mid H_k)P(H_k)}. \quad (10)$$

Fusion cell in this node is a component where the input source data are classified with a convolutional neural network (CNN). CNN is a deep learning method more often applied in image recognition. The aim of the process of CNN is to map the  $n$  input aligned data to  $m$  output health risks in a vast sampling context with high performance (Figure 11). As introduced above, during the learning phase the construction of the approximation function of the neural network is mainly achieved by iteratively adjusting the values of the connection

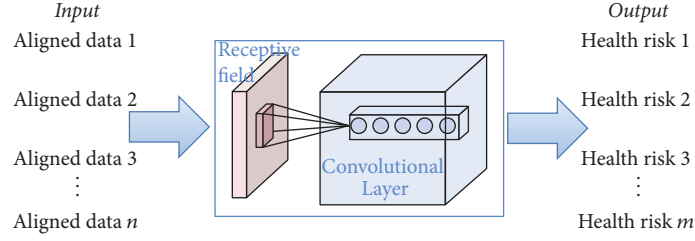


FIGURE 11: CNN-based health risk mapping.

weights of the nodes between different layers. With the optimized weight values, the approximation function is expected to produce outputs with fewer estimation errors. The training error can be calculated by comparing the differences between the outputs of the inference engine and the fusion cell. The calculation of the error can be defined as

$$E = \frac{1}{2N} \sum_{i=1}^N (t_i - z_i)^2, \quad (11)$$

where  $E$  denotes the error,  $t_i$  the target outputs from inference engine,  $z_i$  the predicted outputs from the neural network, and  $N$  the total number of training patterns [30].

The construction of the CNN is similar to the conventional neural networks introduced in the previous section, which are composed of neurons that have learnable weights and biases. Each neuron receives inputs and creates a dot product. The whole network expresses a single differentiate function that scores the input aligned data of health attributes in accordance with the classes of health risks. CNN has been proven to be more efficient in training inputs with a restricted number of parameters and hidden units. CNN can achieve local connections and tied weights efficiently by pooling translation invariant features. This specialty is adaptive to our design, as the input health data has been normalized and the output health risks have been predefined in certain classes [10].

Convolutional layer is the core building block of the network that performs most computations. The parameters of the layer consist of a set of learnable filters. Each filter has a small receptive field. In our context, this can be the subset of the input health documents, by putting the width, height, and depth of the imagery data into a 2D matrix. For acoustic data, we can cast the frequency and utterance of audio to receptive fields [45]. During the forward pass, each filter is convolved across the matrix of the input. The dot product is computed between the filter and the input, and a 2D activation map that represents the responses of the filter at every spatial position is produced. The network learns the activation of the filters when it detects some specific patterns at some spatial position in the input. Along time frames, the activation maps for all filters stack and form the output of the convolution layer.

Pooling layer is commonly inserted between successive convolutional layers in the architecture to progressively reduce the spatial size of the feature maps while retaining essential information retained. It also controls the overfitting of the neural network. The most common form of pooling

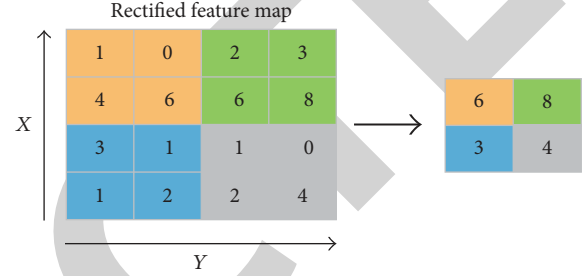


FIGURE 12: Max pooling example.

applies  $2 \times 2$  filters with a stride of 2 at every slice in the input (see Figure 12). The largest elements within each filter are taken from the rectified feature map [46].

After several convolutional and max pooling layers, the high-level reasoning in the neural network is achieved via fully connected layer. Akin to the traditional multilayer perceptron (MLP), the neurons in the fully connected layer have complete connections with all activations in the previous layer. As a result, the activations can be computed with a matrix multiplication followed by a bias offset. The output of the connected layer is the probability distribution over labels, which can be defined as

$$P(y | U) = \frac{\exp(U \cdot w_j)}{\sum_{j=1}^k (U \cdot w_j)}, \quad (12)$$

where  $U$  denotes features produced by the pooling operation  $U = \{c_1, c_2, \dots, c_p\}$  ( $p$  is the number of features),  $k$  is the number of filters, and  $w$  denotes the weight matrix of the fully connected layer [47].

Putting all these together, we have the architecture illustrated as Figure 13. An additional operation called Introducing Nonlinearity, usually achieved via Rectified Linear Unit (ReLU), can be used after every convolution operation, which increases the nonlinear properties of the decision function and of the overall network without affecting the receptive fields of the convolution layer. It computes the function  $f(x) = \max(0, x)$ . In other words, the activation simply has a threshold at zero.

At the third level of the fusion framework, at the time frame where no input probabilities can be provided, the neural network trained in feature integration can produce outputs for state estimation. In a timeframe where health records cannot be provided, a health estimation can be made

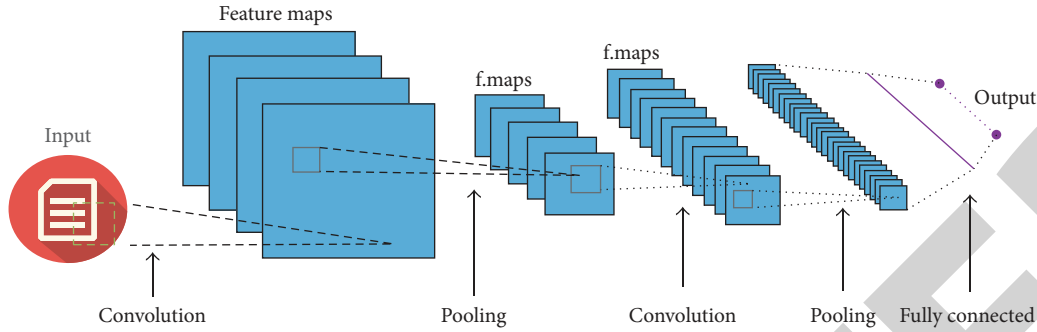


FIGURE 13: Overall CNN architecture.

based on the given inputs of normalized data. Note that the neural network cannot make predictions about future state where no input source data can be obtained. It can only make estimations for the missing previous or current states. With the estimated past and current probabilities of various health risks, time series analyses can be applied to predict the likelihood of future health risks. This process is called target state estimation in the framework. We propose to use ARMA (Autoregressive Moving Average) model for the prediction [48]. The model  $ARMA(p, q)$  can be defined as

$$X_t = \sum_{i=1}^p a_i X_{t-i} + \sum_{i=1}^q b_i w_{t-i} + w_t, \quad (13)$$

where  $a_i$  and  $b_i$  are the weights applied to the model,  $p$  is the autoregressive order,  $q$  is the moving-average order, and  $w_t$  is the white noise.  $ARMA(2, 1)$  is taken frequently to render a weekly stationary process for time series analysis, and the equation can be simplified as

$$X_t = a_1 X_{t-1} + a_2 X_{t-2} + b_1 w_{t-1} + w_t. \quad (14)$$

Fitting it with the past and current probabilities of the health risks, we are able to calculate the weights and white noise of the model. Thereby, the model can be used to predict the probabilities of future health risks.

## 5. Architecture Design and System Implementation

In the above section, we presented a conceptual framework that contains the essential processes for deep learning on eHealth data. This framework provides guidance on building an enhanced health analytic system. The proposed framework forms the basis for an architecture design which we have named Deep Health Analytic Architecture (DHAA). The architecture design is shown in Figure 14. In contrast to the architecture introduced by Wang et al. [6], DHAA applies the concepts of deep learning and information fusion to provide more reliable health prediction. Moreover, the new framework encourages MapReduce and SOA (Service-Oriented Architecture) paradigms to facilitate the distributed computing and systematic communication in enterprise environments.

In correspondence with DHAF, the architecture maps the three-level information fusion processes into four primary components, namely, data alignment, feature extraction, feature integration, and target state estimation. Data alignment consists of fusion nodes that normalize the raw data from different sources. Feature extraction consists of fusion nodes that fuse the normalized data for each data source. Feature integration updates the global situation representation and the estimation model of the neural network with three subcomponents in terms of inference engine, neural network Fusion, and error estimation. Target statement estimation performs health risk prediction based on the neural network model and regression prediction and generates analysis reports. The aggregation of the above four data analytic components are named as Analytic Engine. As an additional component in the architecture, Data Governance provides several functionalities including an input interface for raw data, cache for normalized data, storage for fusion knowledge bases, and access for the analysis reports. Data Governance is the sole interface for all the data to enter and exit the Analytic Engine. This centralized data management eases the utilization of distributed data storage and integrated data access through service calls.

In addition to the architecture, various techniques are applied for system implementation. The operating systems, programming languages, and software frameworks are carefully chosen to optimize the performance of the analysis. As a big data analytic system, the MapReduce paradigm is an important technique useful for enhancing analysis efficiency. MapReduce is a framework with the capability to process large datasets with a parallel, distributed algorithm in a cluster. MapReduce helps to overcome concurrency, robustness, scale, and other common challenges in the analysis for large volume of data [49], as it avoids most low-level implementation problems in distributed computing. As the name suggests, MapReduce mainly consists of a Map step and a Reduce step. Both the Map and Reduce steps can be distributed in a way that takes advantage of the multiple processor cores that are found in modern hardware to increase data process efficiency [50]. In the Map step, the data processing script is executed to produce new key/value pairs. The input data will be grouped on the key, and the value is the information pertinent to the analysis in the Reduce step. In the Reduce step, a function is executed once per key

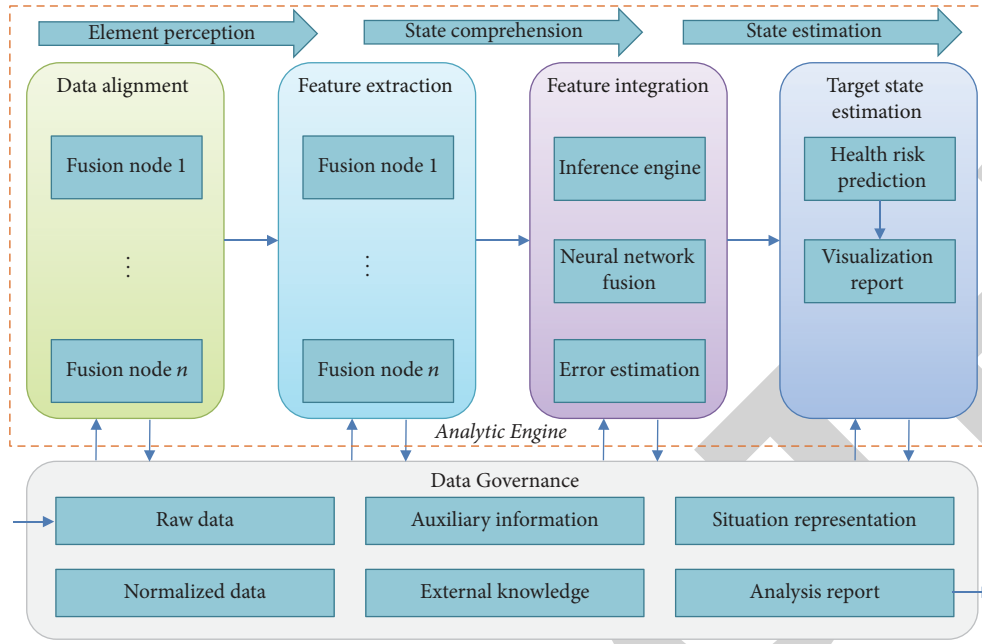


FIGURE 14: Deep Health Analytic Architecture.

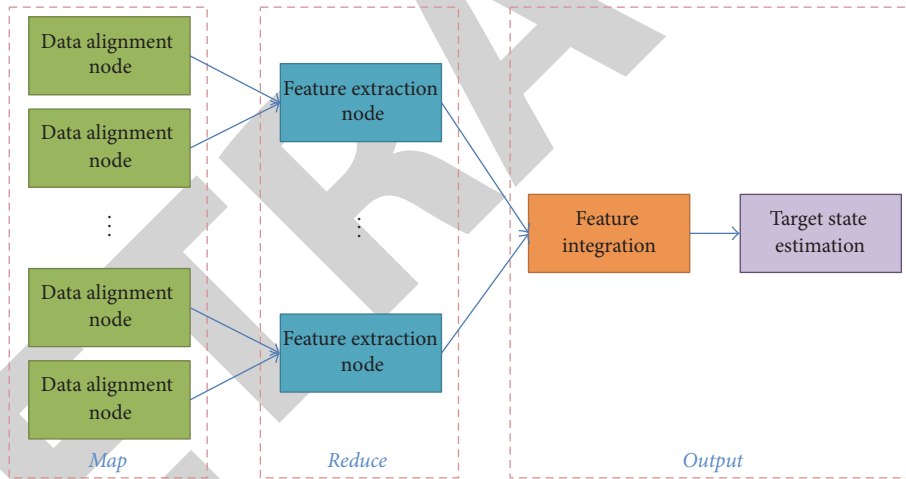


FIGURE 15: DHAA MapReduce pattern.

grouping, and it iterates over all of the values associated with the key. In this function, various tasks can be performed, such as data aggregation and filtering. Once the function is completed, the results in the key/value pair are sent to the Output function.

The Analytic Engine of the DHAA can be fitted into a MapReduce pattern to enhance analysis efficiency. The enhanced pattern is structured as Figure 15.

The data from different sources are processed independently by the nodes of data alignment. These tasks can be performed in parallel to accelerate the processing, and they should be placed in the Map step of the analysis. The aligned data will be aggregated by the nodes of feature extraction based on their data sources. This grouping operation can be achieved by the key/value pairs generated in the Map step.

The processes of feature extraction should be placed in the Map function. The processes of feature integration and target state estimation need to be rendered on the base of the prior two components. Hence, these processes should be placed in the Output function. The pseudocode for the DHAA MapReduce pattern is outlined as shown in Pseudocode 1.

We are implementing a health analytic system based on the above architecture to enhance the health report component of AIA Vitality. We name it Vitality Health Analytic Platform (VHAP, Figure 16). Through the implementation, we can also evaluate the applicability of DHAF form empirics. An experimental system is built on a virtual machine with the Ubuntu Linux operating system and Hadoop, which is a big data-oriented framework that enables distributed storage and distributed processing in computing clusters [51].



```

(1) //set up each data item and encode data in key-value pairs
(2) class mapper:
(3)   setup():
(4)     initialize input element
(5)   map(key,element):
(6)     flag data source to reducer
(7)     identify data type
(8)     normalize data element
(9)   cleanup():
(10)    emit (key,aligned element)
(11) //process key-value pairs and update local model
(12) class reducer:
(13)   setup():
(14)     initialize aligned element
(15)     initialize local situation representations
(16)   reduce(key,elements):
(17)     for element in elements:
(18)       if element meets hypothesis:
(19)         update local situation representation per flag
(20)   cleanup():
(21)     emit (key,local situation representations)
(22) //adjust global model and perform predictions
(23) class outputter:
(24)   setup():
(25)     initialize input elements
(26)     initialize local situation representations
(27)     initialize global situation representation
(28)   output(key,elements,representations):
(29)     for element in input elements:
(30)       if element has proper local situation representation:
(31)         update global situation
(32)         train neural network
(33)         estimate error
(34)         update neural network weights
(35)       else:
(36)         perform neural network prediction
(37)         perform regression prediction
(38)         generate report
(39)   cleanup():
(40)     emit report

```

PSEUDOCODE 1

The Analytic Engine is implemented with R programming language, which provides various libraries convenient for statistical computing and visualization. RHadoop consists of several libraries that allow easy and tight integration between R and Hadoop. Typically, the “rmr2” library contains a collection of functions for implementing MapReduce processing, the “rdfs” library provides interfaces for accessing to HDFS (Hadoop Distributed Filesystem), and the “rhbase” provides an interface for accessing HBase that is a NoSQL database running on top of Hadoop.

There are also numerous R libraries for implementing machine-learning functions. For building the neural networks used in the components of data alignment and feature integration, we can use the “h2o,” “deepnet,” and “MXNetR” libraries. In particular, the “MXNetR” package contains interfaces for implementing convolutional neural networks.

For implementing the Bayesian rules for hypothesis selection in feature extraction, we can use the “OpenBUGS” library. For constructing the Bayesian Networks used in the components of feature extraction and target state estimation, we can use “bnlearn” and “gRain.” For building the regression model for time series analysis in the target state estimation component, we can use the “arima” library. By setting `arima(2,0,1)` in the library, we can perform the ARMA(2,1) analysis. Some sample R scripts for the implementation of machine learnings are provided in Pseudocode 2 as an indication of implementation.

The Data Governance component in VHAP is mainly implemented with the HDFS and HBase of Hadoop. HDFS is used for storing the configuration of external knowledge and auxiliary information. The advantages of using HDFS include the support for files of large size and volume and

```

(1) ## train an neural network for health risk prediction
(2) trainCNN <- function(trainset,risks) {
(3)   require(mxnet);
(4)   # Define the input data
(5)   data <- mx.symbol.Variable('data');
(6)   # 1st convolutional layer
(7)   conv_1 <- mx.symbol.Convolution(data = data, kernel = c(5,5), num_filter = 20);
(8)   act_1 <- mx.symbol.Activation(data = conv_1, act_type = "relu")
(9)   pool_1 <- mx.symbol.Pooling(data = act_1, pool_type = "max", kernel = c(2,2),
      stride = c(2,2))
(10)  # 2nd convolutional layer
(11)  conv_2 <- mx.symbol.Convolution(data = pool_1, kernel = c(5,5), num_filter = 50)
(12)  act_2 <- mx.symbol.Activation(data = conv_2, act_type = "relu")
(13)  pool_2 <- mx.symbol.Pooling(data=act_2, pool_type = "max", c(2,2), stride = c(2,2))
(14)  # 1st fully connected layer
(15)  flatten <- mx.symbol.Flatten(data = pool_2)
(16)  fc_1 <- mx.symbol.FullyConnected(data = flatten, num_hidden = 100)
(17)  act_3 <- mx.symbol.Activation(data = fc_1, act_type = "relu")
(18)  # 2nd fully connected layer
(19)  fc_2 <- mx.symbol.FullyConnected(data = act_3, num_hidden = 40)
(20)  # Set up the symbolic model
(21)  NN_model <- mx.symbol.SoftmaxOutput(data = fc_2)
(22)  # Set seed for reproducibility
(23)  mx.set.seed(0)
(24)  # Train the model
(25)  model <- mx.model.FeedForward.create(NN_model, X=trainset, y=risks, ctx=mx.cpu(),
      num.round = 50, array.batch.size = 40, learning.rate = 0.01, momentum = 0.9)
(26)  return(model);
(27) }

(1) ## use an neural network to predict health risk
(2) predictRisk <- function(nn,dataset,risks) {
(3)   require(neuralnet);
(4)   nn.results <- compute(nn, dataset);
(5)   pr = creditnet.results$net.result;
(6)   colnames(pr) <- risks;
(7)   return(pr);
(8) }

(1) ## use ARMA to predict health risk and plot
(2) plotARMA <- function(dataset,start=c(2011,1)) {
(3)   require(forecast);
(4)   datat<-ts(dataset[,2],start=start,frequency=12);
(5)   fit <- Arima(datat, order=c(2,0,1));
(6)   plot(forecast(fit));
(7) }

```

## PSEUDOCODE 2

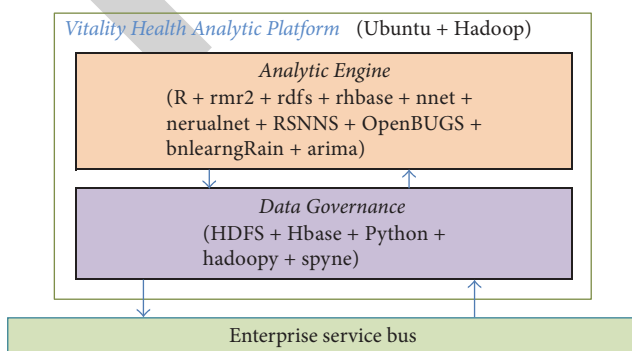


FIGURE 16: Vitality Health Analytic Platform.

high efficiency for frequent data access. HBase is used for caching the raw data and aligned data, managing the state representations, and keeping the analysis reports. In contrast to traditional databases, HBase provides a fault-tolerant way of storing large quantities of sparse data. The communication between Data Governance and the external system is through the web services implemented by Python, which is a programming language with reliable Hadoop integration and handy network communication supports. The “hadoopy” library enables Python to interact with Hadoop. The “spyne” library provides a framework for implementing and exposing remote procedure-call APIs. These APIs are exposed to the enterprise service bus, and Data Governance becomes



FIGURE 17: New health report versus old health report.

the interface between the Analytic Engine and the external systems.

In comparison with the previous health reports in AIA Vitality (see Figure 17), the new health reports using VHAP can provide health risk predictions with higher granularity. While the previous health reports based on data rules can only provide health predictions with broadly indicative predictions, the new prediction can give indicators of health risks on an annual basis and can predict future trends for each health risk.

In order to verify the performance of the selection of CNN for fusion cell implementation, we substituted the implementation of CNN with other learning techniques in terms of SVM, KNN, and LR for comparison. The health records of approximately 20,000 AIA Vitality members were sampled for the calculation. The number of the original records of the sample members was about 800,000,000. However, there were many duplicated records which are produced from their source systems. For example, Vitality Mobile App users tend to upload the same data several times during a day. On the other hand, invalid records also exist, likely due to the malfunction of the source systems or the inappropriate operations of the software by users. The data quality needs to be controlled to ensure the accuracy and performance of the analytics. This is achieved by implementing a filter in the data alignment node, which screens out the duplicated and outlier data before further processes are applied. The classification accuracy and kappa measure for the specific implementations are given in Figure 18, and their efficiency measures are given in Figure 19.

From Figure 18, it can be seen that CNN has relatively high accuracy compared to other learning methods. However, during the experiments, we noticed that the process duration of CNN was longer than other methods. We believe that is caused by the complex implementation of CNN. In contrast, KNN is a simpler classification algorithm. While its overall accuracy is acceptable, its accuracy is usually not stabilized depending on the shape of the dataset. LR is even

more sensitive to the distribution of the sample data. Outliers in the sample data can dramatically affect the result. Both SVM and CNN are state-of-the-art learning algorithms. They both learn from the inputs and training data by adjusting their weights. SVM achieves the weighting by mapping the inputs to higher dimensional space, while CNN achieves it by producing multilayer feature maps. The complex architecture of CNN makes it a stronger classifier and its results are more reliable.

According to Figure 19, CNN has the longest process duration and lowest throughput when performing the sample analytics, whereas LR and KNN require much less processing time and have high throughput when analyzing the sample data. However, taking into account the accuracy figures, although CNN consumes more process time, it produces higher accuracy. In contrast, LR requires less processing time and its accuracy is also low. In addition, the process duration can be reduced if we increase the capacity of the platform. Nevertheless, it is more difficult to increase the accuracy of analytics than enhance the computing capacity.

## 6. Conclusions

This research explores the potential value for utilizing the concepts of the fusion node paradigm and deep learning to enhance health risk prediction from big data. The fusion node paradigm improves the structure and workflow of the health analytic design, so that the analytic processes can be conducted in a more efficient way. Deep learning compounds multiple machine-learning methods in complex processes to enhance data analysis performance. Using deep learning techniques in the analytic processes allows health risks to be predicted in a more reliable manner, as it allows iterative inference of high-level information based on low-level data. In addition, deep learning methods such as convolutional neural network have the potential to enhance analysis accuracy with large training sets. This specialism has great merit in this era of big data. A framework is proposed based on

Risks	SVM	KNN	LR	CNN
Obesity	53.89	84.98	35.61	71.43
Diabetes	79.06	65.65	44.87	96.28
Stroke	65.77	67.55	84.56	47.36
Heart disease	73.43	84.09	23.56	72.81
High blood pressure	68.51	52.48	61.13	75.42
Osteoarthritis	77.57	65.31	67.38	75.35
Cancer	60.09	55.46	34.74	80.47
Overall	67.41	65.32	48.86	73.21
Kappa	0.59	0.57	0.41	0.63

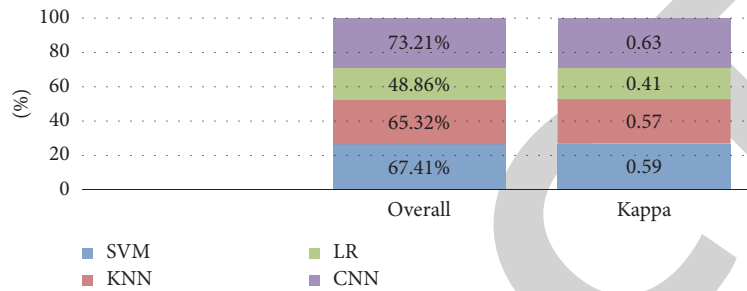


FIGURE 18: Accuracy and kappa of various implementations.

	SVM	KNN	LR	CNN
Proc duration (Min)	1247	893	694	1652
Throughput (rows/Sec)	4147	5725	7359	3202
Throughput (KB/Sec)	3390	4644	5944	2610

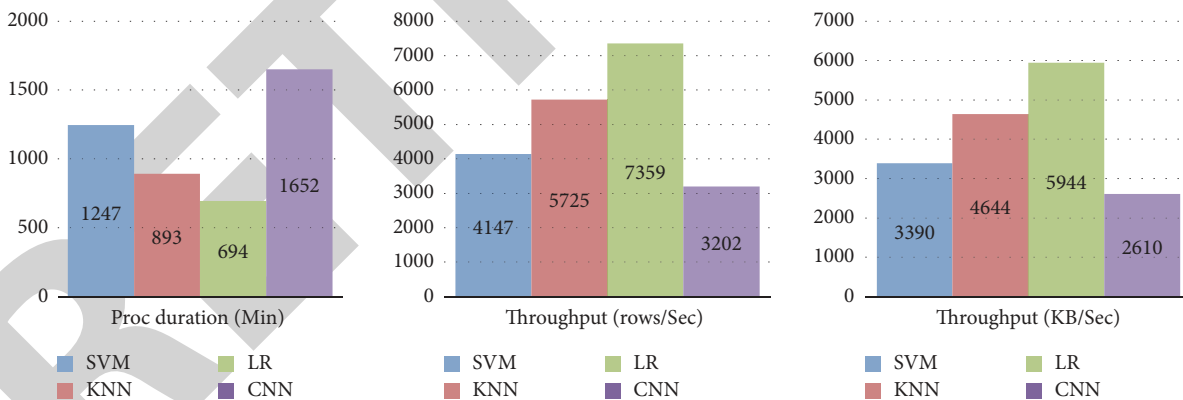


FIGURE 19: Efficiency of various implementations.

the concepts described above to discuss the theories and processes of the analytic workflow and its components. An architecture is designed based on the proposed framework to illustrate how to organize the analytic components of the analytic system to capitalize on the cohesion and coupling. This paper also discussed the application of the MapReduce paradigm to optimize the efficiency of the analytic processes in the architecture. Finally, we encourage SOA design of the analytic system to ease the ability of external systems to access the analytic results.

In the future, we plan to perform more analysis on the performance of the proposed prediction framework. More observations are required to verify whether the predicted probabilities are aligned with the actual occurrence rates. Besides the accuracy, computation efficiency is another concern of analytic systems. There need to be an observation of the process duration and resource consumption and a comparison of different algorithms in regard to the analytic processes. We can also attempt to devise different mathematic models for probability aggregation.



## Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

## References

- [1] J. Wang, Z.-Q. Zhao, X. Hu, Y.-M. Cheung, H. Hu, and F. Gu, "Online learning towards big data analysis in health informatics," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 8211, pp. 516–523, 2013.
- [2] W. Liu and E. Park, "Big data as an e-health service," in *Proceedings of the 2014 International Conference on Computing, Networking and Communications ICNC 2014*, pp. 982–988, February 2014.
- [3] E. Mezghani, E. Exposito, K. Drira, M. Da Silveira, and C. Pruski, "A Semantic Big Data Platform for Integrating Heterogeneous Wearable Data in Healthcare," *Journal of Medical Systems*, vol. 39, no. 12, article no. 185, 2015.
- [4] L. Deng and D. Yu, "Deep learning: Methods and applications," *Foundations and Trends in Signal Processing*, vol. 7, no. 3-4, pp. 197–387, 2013.
- [5] H. Chai and B. Wang, "A hierarchical situation assessment model based on fuzzy Bayesian network," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 7003, no. 2, pp. 444–454, 2011.
- [6] Y. Wang, L. Kung, and T. A. Byrd, "Big data analytics: understanding its capabilities and potential benefits for healthcare organizations," *Technological Forecasting & Social Change*, pp. 1–11, 2016.
- [7] N. Jothi, N. A. Rashid, and W. Husain, "Data Mining in Healthcare - A Review," in *Proceedings of the 3rd Information Systems International Conference*, pp. 306–313, April 2015.
- [8] S. Theodoridis, *Machine Learning - A Bayesian and Optimization Perspective*, Elsevier, 2015.
- [9] S. Zillner and S. Neururer, "Big data in the health sector," in *New Horizons for a Data-Driven Economy*, pp. 179–194, Springer International Publishing.
- [10] I. Goodfellow and A. Courville, *Deep learning*, MIT Press, 2016.
- [11] Z. Liang, G. Zhang, J. X. Huang, and Q. V. Hu, "Deep learning for healthcare decision making with EMRs," in *Proceedings of the IEEE International Conference on Bioinformatics and Biomedicine*, pp. 556–559, November 2014.
- [12] E. Bosse, J. Roy, and S. Wark, *Concepts, Models, and Tools for Information Fusion*, Artech House, 2007.
- [13] X. Gao, W. Lia, M. Loomesa, and L. Wang, "A fused deep learning architecture for viewpoint classification of echocardiography," *Information Fusion*, vol. 36, pp. 103–113, 2017.
- [14] NIST Big Data Public Working Group, "NIST Big Data Interoperability Framework," 2015, <http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.1500-4.pdf>.
- [15] Y. Wang, L. Kung, C. Ting, and T. A. Byrd, "Beyond a technical perspective: Understanding big data capabilities in health care," in *Proceedings of the 48th Hawaii International Conference on System Sciences*, pp. 3044–3053, January 2015.
- [16] B. Lantz, *Machine Learning with R 2ed*, Packt Publishing, 2015.
- [17] J. Bell, *Machine Learning - Hands-On for Developers and Technical Professionals*, John Wiley & Sons, 2015.
- [18] E. Vermeulen-Smit, M. Ten Have, M. Van Laar, and R. De Graaf, "Clustering of health risk behaviours and the relationship with mental disorders," *Journal of Affective Disorders*, vol. 171, pp. 111–119, 2015.
- [19] D. Talia, D. Talia, and F. Marozzo, *Data Analysis in the Cloud - Models, Techniques and Applications*, Elsevier Inc, 2016.
- [20] J. Kazmierska and J. Malicki, "Application of the Naïve Bayesian Classifier to optimize treatment decisions," *Radiotherapy and Oncology*, vol. 86, no. 2, pp. 211–216, 2008.
- [21] P. Harrington, "Machine learning in action," *Manning Publications*, 2012.
- [22] J. Sander and J. Beyerer, "Bayesian fusion: Modeling and application," in *Proceedings of the Sensor Data Fusion: Trends, Solutions, Applications*, pp. 1–6, October 2013.
- [23] L. A. Klein, *Sensor and Data Fusion: A Tool for Information Assessment and Decision Making*, SPIE Press, 2004.
- [24] A. Gelman, H. S. Stern, D. B. Dunson, A. Vehtari, and D. B. Rubin, *Bayesian Data Analysis*, CRC Press, 3rd edition, 2014.
- [25] N. Fenton and M. Neil, "Decision support software for probabilistic risk assessment using bayesian networks," *IEEE Software*, vol. 31, no. 2, pp. 21–26, 2014.
- [26] M. L. Vaneek, "Introduction into Bayesian Networks," 2016, <http://www.fit.vutbr.cz/study/courses/VPD/public/0809VPD-Vaneek.pdf>.
- [27] D. Barber, *Bayesian Reasoning and Machine Learning*, Cambridge University Press, 2012.
- [28] D. E. Holmes and L. C. Jain, *Innovations in Bayesian Networks*, Springer, 2008.
- [29] A. Exposito, S. Bassis, and F. C. Morabito, "Recent advances of neural networks models and applications: An introduction," *Smart Innovation, Systems and Technologies*, vol. 37, pp. 3–8, 2015.
- [30] S. Samarasinghe, *Neural Network for Applied Sciences and Engineering*, Auerbach Publications, 2007.
- [31] M. Kirk, *Thoughtful Machine Learning*, O'Reilly Media.
- [32] N. Lewis, *Deep Learning Made Easy with R*, CreateSpace, 2016.
- [33] N. Dhungela, G. Carneiro, and A. P. Bradley, "A deep learning approach for the analysis of masses in mammograms with minimal user intervention," *Medical Image Analysis*, vol. 37, pp. 114–128, 2017.
- [34] M. Perez, S. Avilab, D. Moreiraa et al. et al., "Video pornography detection through deep learning techniques and motion information," *Neurocomputing*, vol. 230, pp. 279–293, 2017.
- [35] Q. Zhang, L. T. Yang, and Z. Chen, "Privacy preserving deep computation model on cloud for big data feature learning," *Institute of Electrical and Electronics Engineers. Transactions on Computers*, vol. 65, no. 5, pp. 1351–1362, 2016.
- [36] S. Das, *High-Level Data Fusion*, Artech House, 2008.
- [37] M. K. Sundareshan and Y. C. Wong, "Nueal network-based fusion of image and non-image data for surveillance and tracking application," *Data Fusion for Situation Monitoring, Incident Detection, Alert and Response Management*, no. IOS Press, 2015.
- [38] H. B. Mitchell, "Data fusion: Concepts and ideas," *Data Fusion: Concepts and Ideas*, pp. 1–344, 2012.
- [39] D. L. Hall and J. Llinas, *Handbook of Multisensor Data Fusion*, CRC Press, 2001.
- [40] E. Bosse and B. Solaiman, *Information Fusion and Analytics for Big Data and IoT*, Artech House, 2016.

- [41] L. Wald, *Data Fusion: Definitions and Architectures*, Les Presses de L'Ecole des Mines, 2002.
- [42] M. C. Amirani, M. Toorani, and A. A. Beheshti, "A new approach to content-based file type detection," in *Proceedings of the 13th IEEE Symposium on Computers and Communications, ISCC 2008*, pp. 1103–1108, July 2008.
- [43] D. Tian, A. Gledson, A. Antoniadis, A. Aristodimou, and N. Dimitrios, "A bayesian association rule mining algorithm," in *Proceedings of the IEEE International Conference on Systems*, pp. 3258–3264, October 2013.
- [44] D. Allard, A. Comunian, and P. Renard, "Probability aggregation methods in geoscience," *Mathematical Geosciences*, vol. 44, no. 5, pp. 545–581, 2012.
- [45] O. Abdel-Hamid, A.-R. Mohamed, H. Jiang, L. Deng, G. Penn, and D. Yu, "Convolutional neural networks for speech recognition," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 22, no. 10, pp. 1533–1545, 2014.
- [46] M. Sun, Z. Song, X. Jiang, J. Pan, and Y. Pang, "Learning Pooling for Convolutional Neural Network," *Neurocomputing*, vol. 224, pp. 96–104, 2017.
- [47] Q. Li, Z. Jin, C. Wang, and D. D. Zeng, "Mining opinion summarizations using convolutional neural networks in Chinese microblogging systems," *Knowledge-Based Systems*, vol. 107, pp. 289–300, 2016.
- [48] R. Prado and M. West, *Time Series - Modeling, Computation, and Inference*, CRC Press, 2010.
- [49] D. Miner and A. Shook, *MapReduce Design Patterns*, O'Reilly Media, 2013.
- [50] B. Holt, *Writing and Querying MapReduce Views in CouchDB*, O'Reilly Media, 2011.
- [51] V. Prajapati, *Big Data Analytics with R and Hadoop*, Packt Publishing, 2013.