International Joint Conference on Neural Networks, 2017 (IJCNN 2017), 14-19 May 2017, Anchorage, Alaska. USA

# A New Semantic Attribute Deep Learning with a Linguistic Attribute Hierarchy for Spam Detection

Hongmei He*, Tim Watson †, Carsten Maple †, Jörn Mehnen‡, Ashutosh Tiwari*,
*Manufacturing Informatics Centre, Cranfield University, Cranfield, MK43 0AL, UK
Email: h.he@cranifeld.ac.uk, a.tiwari@cranfield.ac.uk
‡Cyber Security Centre - WMG, University of Warwick, UK
Email: tw@warwick.ac.uk, cm@warwick.ac.uk
†Design, Manufacture & Engineering Management, University of Strathclyde, Glasgow, G1 1XQ, UK
jorn.mehnen@strath.ac.uk

***Abstract*—The massive increase of spam is posing a very serious threat to email and SMS, which have become an important means of communication. Not only do spams annoy users, but they also become a security threat. Machine learning techniques have been widely used for spam detection. In this paper, we propose another form of deep learning, a linguistic attribute hierarchy, embedded with linguistic decision trees, for spam detection, and examine the effect of semantic attributes on the spam detection, represented by the linguistic attribute hierarchy. A case study on the SMS message database from the UCI machine learning repository has shown that a linguistic attribute hierarchy embedded with linguistic decision trees provides a transparent approach to in-depth analysing attribute impact on spam detection. This approach can not only efficiently tackle 'curse of dimensionality' in spam detection with massive attributes, but also improve the performance of spam detection when the semantic attributes are constructed to a proper hierarchy.**

## I. Introduction

Spam is an ever-increasing problem. It pervades any information system through email or web, social, blog or reviews platform [1], and is increasingly being used to distribute virus, spyware, links to phishing web sites, etc. Email spam detection has been an important part of correspondence since email became an essential part of our daily lives. The growth of of the number of mobile phone users has led to a dramatic increasing of SMS spam messages [2], and message spams have increasingly disturbed human life and caused significant economic losses to many mobile users. The problem of spam is not only an annoyance, but has also become a security threat. It has attracted much attention of researchers for decades.

Web spam can be categorised into content spam (overly stuffed and badly rated keywords), link spam (outgoing link spam, incoming link spam), cloaking & redirection, and click spam. Content spam is probably the first and most widespread form of web spam [1]. PageRank [3] is the most classic algorithm to estimate the global importance (authority or reputation) score of a webpage on the web, which was used by Google Search. TrustRank [4], Anti-Trust Rank [5] and SpamRank [6] are well-known link-based detection algorithms. It was shown that machine learning is superior to the PageRank algorithm for static page ranking [7]. Silva et al. [8] compared most classical machine learning methods, such as Artificial Neural Networks (ANN), Support Vector Machine (SVM), Decision Tree (DT), Random Forest (RF), K-Nearest Neighbors (KNN), Adaptive Boosting, Bagging and LogitBoost for web spams. Recently, Wang et al. [9] investigated social spam detection on Twitter with Bernoulli Naive Bayes (NB), KNN, SVM, DT and RF, and the RF algorithm obtained the best $F_1$-measure up to 0.946 on the social Honeypot dataset. Crawford et al. [10] applied machine learning techniques to detect false reviews (e.g. making fake, untruthful, or deceptive reviews), of which the purpose is to artificially promote or devalue products and services for profit or gain. Verma and Dhawan [12] investigated spam detection in social networks with clustered KNN technique.

Email spam could also have content spam and click spam. While web spams usually target public users or some specific user groups, email spams may have clearer targets for specific purposes. Much research on email spam detection is based on content spam. Machine learning technologies are widely applied for email spam detection. The family of NB classifiers [13], [14] is one of the most commonly implemented, which has been embedded in many popular email clients. Tretyakov [15] used the combination of the most classic machine learning methods (Bayesian classifier, KNN, ANNs, SVMs) for the problem of email spam-filtering. The combination of Bayesian classifier and SVM obtained the precision of 94.4%. Shirani-Mehr [11] investigated SMS Spams with NB algorithm.

Feature extraction is also important in spam detection. Wang et al. [9] used four different types of feature sets (user features, content features, Uni-Bi features, and sentiment features) and their combinations to validate Random Forests for Twitter spam detection, and the experimental results show feature combination outperformed a single type of features on decision making. The study of Alqatawna et al. [16] showed that adding malicious related features to training data significantly improved the detection of spam emails. Recently, He et al. [17] investigated spam detection through analysing the features in email and message spams with a RBF-kernal SVM spam detector, thus to provide clues of spams to users.

Deep learning is a branch of machine learning based on a set of algorithms that attempt to model high level abstractions in data by using a deep graph with multiple processing layers of nonlinear processing units for feature extraction and transformation. Each successive layer uses the output from the previous layer as input, forming a hierarchy from low-level to high-level features. A linguistic attribute hierarchy has similar

properties to the generic deep learning. Intermediate attributes in upper layer are functions of their child features (attributes) in lower layer. Hence, a linguistic attribute hierarchy (LAH) can be viewed as another form of deep learning. Unlike the generic deep learning, which constructs a neural network architecture, such a linguistic attribute hierarchy has a flexible tree structure with all attributes as leaves, and all non-leaf intermediate nodes and the top decision node of the tree can be represented by a machine learning model (e.g. linguistic decision tree [18], [19] and neural network [20]), fed with their child nodes. An LAH, embedded with LDTs, integrates the linguistic rules of LDTs in the hierarchy to form a hierarchical decision making or classification. The linguistic rules in the hierarchy are produced by the embedded LDTs (equivalent to the nonlinear process units in generic deep learning) in terms of hierarchical process of decision making or classification (see details in Section II.C).

The 'curse of dimensionality' in decision making modelling is the exponentially increasing number of rules associated with the increasing number of input attributes. This is a major bottleneck that blocks the applications of decision trees. With $n$ input variables, described with $\tau$ distinguishable labels, a full decision tree could produce $\tau^n$ number of rules. The research [18] has shown that a linguistic attribute hierarchy could help tackling the 'curse of dimensionality' problem, and improve the performance of decision making or classification.

In this paper, a linguistic attribute hierarchy (LAH), embedded with LDTs, is investigated for the deep learning of attribute semantics for spam detection. The set of features extracted from the message spam database is decomposed to subsets of features in terms of the attribute semantics. Each subset will be fed to an LDT to form an intermediate or the final decision varialbes in an LAH, and an LDT in upper level of the LAH could be fed with a subset of features and/or some intermediate attributes from lower level. A case study is performed on the SMS message spam database from the UCI machine learning repository.

## II. METHODOLOGY

### A. Linguistic Decision Tree

*1) Label Semantics:* Label Semantics [21] uses a finite set of successive labels $\mathscr{L}$ to describe an object in an underlying domain $\Omega$. Each label corresponds to a fuzzy interval. Appropriateness measure and mass assignment are two fundamental and interrelated measures in Label Semantics. The appropriateness measure ($\mu_L(x)$) of a label $L$, describing an instance $x$, quantifies the agent's subjective belief how $L$ can be used to describe $x$ based on its (partial) knowledge of the current labeling conventions of the population.

A mass assignment $m_x$ on sets of labels quantifies an agent's belief that any particular subset (called focal element) of labels contains all and only the labels with which it is appropriate to describe $x$. For an object $x \in \Omega$, mass assignment on labels is a function $m_x : 2^{\mathscr{L}} \to [0,1]$, st. $\sum_{S \subseteq \mathscr{L}} m_x(S) = 1$, $S \neq \phi$, as an instance can be described with at least one label, $L \in \mathscr{L}$. In labeling conventions, some combinations of labels may not be appropriate to describe any object. For example, given label set $\mathscr{L} = \{small, medium, large\}$, $\{small, medium\}$ and $\{medium, large\}$ could be used to describe an object, but *small* and *large* cannot both be appropriate to describe an object.

Given a label set $\mathscr{L}$ together with associated mass assignment $m_x$, the set of focal elements for $\mathscr{L}$ is given by: $\mathscr{F} = \{S \subseteq \mathscr{L} : \exists m_x(S) > 0\}$.

A $\lambda$-mapping of a label $L \in \mathscr{L}$ is defined as $\lambda(L) = \{F \in \mathscr{F}, L \in F\}$. That is, $\lambda(L)$ covers those focal elements that include label $L$.

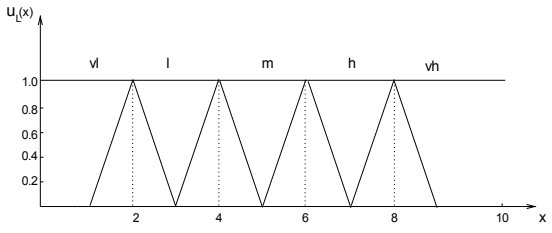
An appropriateness measure $\mu_L(x)$ quantifies the degree of our belief that the label $L$ is appropriate for $x \in \Omega$. The value of $\mu_L(x)$ is evaluated as the sum of mass assignments $m_x$ over $\lambda(L)$:

$$\mu_L(x) = \sum_{F \in \lambda(L)} m_x(F). \tag{1}$$

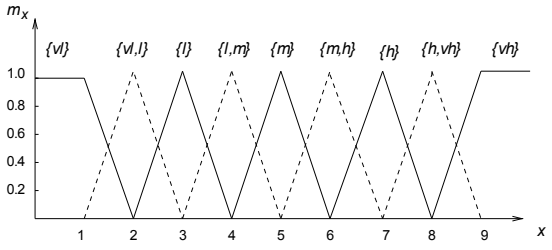
In Label Semantics, the consonance assumption provides an approach to calculating mass assignments from appropriateness measures uniquely. Given non-zero appropriateness measure on basic labels $\mathscr{L} = \{L_1, L_2, ..., L_k\}$ ordered such that $\mu_{L_i}(x) \geq \mu_{L_{i+1}}(x)$, for $i = 1, ..., k-1$, then the consonant mass assignment has the form:

$$m_x(\{L_1, ..., L_k\}) = \mu_{L_k}(x),$$
$$m_x(\{L_1, ..., L_i\}) = \mu_{L_i}(x) - \mu_{L_{i+1}}(x) \ for \ i = 1, ..., k-1.$$

Figs. 1 (a) and (b) depict the relationship between appropriateness measure and mass assignment of focal elements. Fig. 1 (a) shows the appropriateness of attribute $x \in [0, 10]$ on the label set $\mathscr{L} = \{vl, l, m, h, vh\}$. Each label represents an interval, and they overlap by 50%. Fig. 1 (b) presents the mass assignments on focal elements, corresponding to the appropriateness of $x$ on labels in Fig. 1 (a). Here, an attribute $x \in \Omega$ is uniformly divided with same intervals, and there is no empty set on labels. Label learning on data is another topic. For example, in Figure 1 (a), given an instance,




(a) Appropriateness




(b) Mass assignments

Fig. 1. The two fundamental and interrelated measures in Label Semantics

$x \in [2,3]$, $\mu_l(x) = 1$ and $\mu_{vl}(x) = 3 - x$. In terms of the

Consonance Assumption, the order of labels are $\{l, vl\}$. Hence, we have (see Figure 1 (b)): $m_x(\{vl, l\}) = m_x(\{l, vl\}) = 3 - x$ and $m_x(\{l\}) = 1 - (3 - x) = x - 2$, $x \in [2, 3]$.

*2) The Definition of an LDT:* An LDT is comprised of a set of nodes, representing attributes, and edges, representing focal elements describing attributes associated to nodes. A branch ($B$) is a path from the root to a leaf of the LDT, and its length ($l$) is the number of focal elements $F_i$ on the branch $B$, for $i = 1, ...l$. Each branch represents a conjunction of focal elements: $F_1 \wedge ... \wedge F_l$, it is augmented with a set of conditional mass assignments $m(F|B)$, for each focal element $F \in \mathscr{F}_y$, where, $\mathscr{F}_y$ is the set of labels describing the goal variable $y$. Definition 2.1 provides the linguistic definition of an LDT.

*Definition 2.1 (Linguistic Definition):* In an LDT, the rule of the branch $B_i$ is $F_{i_1} \wedge ... \wedge F_{i_l} \rightarrow F : m(F|B_i)$, for each focal element $F \in \mathscr{F}_y$, the mass assignment $m_y$, for a given example with attribute values $\vec{x} = (x_1, \ldots, x_m)$, can be calculated in terms of Jeffrey's rule [22] by

$$m_y(F) = \sum_{i=1}^{t} \mu_{B_i}(\vec{x}) m(F|B_i), \tag{2}$$

where $t$ is the number of branches in the LDT, and $m(F|B_i)$ is equivalent to the conditional probability $p(F|B_i)$. Assume $x_{i_j}$ is expressed with a focal element $F_{i_j}$ in the branch $B_i$. For a branch $B_i$, we have

$$\mu_{B_i}(\vec{x}) = \prod_{j=1}^{l} m_{x_{i_j}}(F_{i_j}). \tag{3}$$

### B. The LID3 algorithm to train an LDT

LID3 [23], [18], an extension of the well-known ID3 algorithm [24], is used to train an LDT based on a given database. The induction is guided by a modified measure of information gain according to label semantics.

*Definition 2.2 (Information Entropy of Branch):* Given a goal variable, belonging to class set $C = \{C_1, ..., C_t\}$, the information entropy of branch $B$ is defined as

$$E(B) = -\sum_{i=1}^{t} P(C_i|B) log_2 P(C_i|B). \tag{4}$$

*Definition 2.3 (Expected Entropy):* When $x_j$ is appended to the branch $B$, the Expected Entropy can be calculated as

$$EE(B, x_j) = \sum_{F_j \in \mathscr{F}_j} E(B \cup F_j) P(F_j|B), \tag{5}$$

where $B \cup F_j$ represents the new branch obtained by appending the focal element $F_j$ to the end of the branch $B$. The probability of $F_j$ given $B$ can be calculated as

$$P(F_j|B) = \frac{\sum_{\vec{x} \in \mathscr{D}} P(B \cup F_j|\vec{x})}{\sum_{\vec{x} \in \mathscr{D}} P(B|\vec{x})}, \tag{6}$$

where $P(B|\vec{x}) = \mu_B(\vec{x})$ (see Definition 2.1).

*Definition 2.4 (Information Gain):* Using the notations of Definitions 2.2 and 2.3, the information gain can be calculated as

$$IG(B, x_j) = E(B) - EE(B, x_j). \tag{7}$$

In LID3, the most informative attribute is selected as the root of an LDT, and the tree will be expanded with branches associated with all possible focal elements of this attribute. For each branch, the attribute with maximal information gain in all free attributes will be the next node until the branch reaches the specified maximum length or the maximum class probability reaches the given threshold. The learning process forms a level order traversal based on the training data.

### C. A hierarchy of LDTs

The process of aggregation of evidence in multi-attribute decision making or classification based on attributes $x_1, ..., x_n$ can be represented with a functional mapping $y = f(x_1, ..., x_n)$, where $y$ is the goal variable. However, this mapping is often dynamic and it is difficult to define it with an exact mathematic equation. An attribute hierarchy decomposes the function $f$ into a hierarchy of sub-functions, each of which represents a new intermediate attribute. The set of original attributes $\{x_1, ..., x_n\}$ is divided into $m$ subsets $S_1, ..., S_m$, and correspondingly $m$ new intermediate attributes $z_1, ..., z_m$ are represented by the functions of the subsets. Namely, $z_i = G_i(S_i)$ for $i = 1, ..., m$. The mapping function $f$ is then equivalent to a new function $F$ of the intermediate attributes $z_1, .., z_m$. Hence, $y = f(x_1, ..., x_n)$ $= F(z_1, ..., z_2) = F(G_1(S_1), ..., G_m(S_m))$. The function of a subset can be decomposed recursively.

Assume each attribute is described with a set of labels, and subsequent label expressions are derived. In a linguistic attribute hierarchy (LAH), functional mappings between parent and child attribute nodes can be defined in terms of weighted linguistic rules that explicitly model both the uncertainty and vagueness, which often indicates our knowledge of such aggregation functions. Now, assume that the introduced intermediate attributes are approximations of the goal variable $y$ with the same domain and description labels. For an example, in Fig. 2, $z_1$ is an approximation of $y$ based only on $x_1$ and $x_2$. Unlike general deep learning, a neural network is constructed through layer by layer training, the new type of deep learning will construct a tree of LDTs. Hence, an LAH can present a hierarchical decision making or classification. With this property, it is easy to perform semantic attribute deep learning through a hierarchy of LDTs.

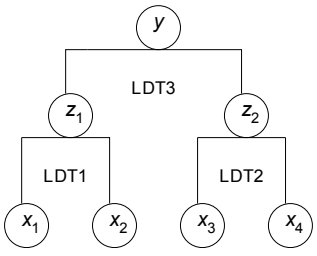



Fig. 2. An example of LAH

## III. Case Study

Here, we demonstrate the use of the proposed approach on a benchmark database from UCI machine learning repository [25], and evaluate the performance of different linguistic attribute hierarchies embedded with LDTs, and compare comparing them with the performance of the single LDT.

### A. The data base

The SMSSpamCollection database, created by Almeida et al. [2], provides 5574 raw messages, of which 747 are spams, accounting for 13.4% of total messages. As SMS messages are short, the features extracted from the SMS message data are binary values. If a KEY-WORD or a kind of behaviour exists in an SMS message, the corresponding feature is set to one, otherwise, it is set to zero, and key-words are not case-sensitive. For example, key-word, 'free' indicates if any words, in which 'free' is partial phase (e.g. Free, FREE, Freedom, free), then $x_1$ will be set to 1, otherwise, 0. $x_{17}$ indicates all spams of noisy advertisements, which claim cheap service price, for instances, 1p/MIN, 1.5p/MIN, ... 2.5p/call, etc; $x_{19}$ intends to find SMS messages, which include money values with different currency units.

The feature set is basically the same as in [17], however, there are some changes in the way features are extracted: (1) the features 'half price' has been removed, as the sample with positive feature 'price' covers all samples with positive feature 'half price'; (2)according to the results in [17], the feature '!' is extracted, as its importance has been shown in the email spam detection; (3) features 'WIN' and 'WON' are combined as one feature by the excel command, 'IF(OR(x1,x2),1,0)'. There are totally 20 features, shown in Table I.

TABLE I. FEATURES FOR THE SMS MESSAGE DATA BASED ON [17]

| $x$ | key word | $x$ | key-word |
|---|---|---|---|
| 0 | urgent | 10 | stop |
| 1 | congrat | 11 | click |
| 2 | ! | 12 | Text,Txt |
| 3 | WIN/WON | 13 | sex |
| 4 | Offer | 14 | girl |
| 5 | Award | 15 | cash |
| 6 | Prize | 16 | free |
| 7 | Call | 17 | 0p, 1p, ..., 9p |
| 8 | Reply | 18 | EURO, GBP, pound,$\mathscr{L}$, \$, € |
| 9 | Send | 19 | price |

### B. Experiment methodologies

**Mass assignments of attributes:** As all features extracted from the SMSSpamCollection database are binary variables, the mass assignment on a discrete label is 1 if the attribute value is 1, otherwise 0.

**Two-fold cross validation:** Data is split into two approximate equal partitions. One is used for training, and the other one is for testing, and reverse the process.

**Performance measure:** Four performance measures will be evaluated:

(1) True positive rate (TPR, sensitivity, recall) measures the proportion of positives that are correctly identified as such. Therefore, sensitivity quantifies the avoiding of false negatives. It is calculated with $TP/P$, where, $TP$ is the number of spams that are correctly identified, and $P$ is the number of spams in the database.

(2) True negative rate (TNR, specificity) measures the proportion of negatives that are correctly identified as such. Therefore, specificity quantifies the avoiding of false positives. It is calculated with $TN/N$, where, $TN$ is the number of hams that are correctly identified, and $N$ is the number of hams in the database.

(3) General accuracy (A) measures the proportion of samples (either spams or hams) that are correctly identified as such. It is calculated with $(TP+TN)/(P+N)$.

(4) ROC curve and Area under ROC curve ($A_{ROC}$): In statistics, a receiver operating characteristic (ROC), or ROC curve, is a graphical plot that illustrates the performance of a binary classifier system as its discrimination threshold is varied. The curve is created by plotting the true positive rate (TPR) against the false positive rate (FPR) at various threshold settings.

### C. Attribute Decomposition

**Decomposition 1** All features could represent some meanings in a message, for example, interjection, attraction, behaviour, benefit, etc. Hence, the feature set extracted from the SMSSpamCollection is decomposed to five subsets semantically in terms of their specific meanings.

(1) Interjection (3): $x_0$:'urgent',$x_1$:'congrat', $x_2$:'!'

(2) Attraction (4): $x_3$:'WIN/WON', $x_4$:'offer', $x_5$:'Award', $x_6$:'Prize'

(3) Behaviour (6): $x_7$:'call', $x_8$:'Reply', $x_9$:'send', $x_{10}$:'stop', $x_{11}$:'click', $x_{12}$:'text/txt'

(4) Benefit (5): $x_{15}$:'cash', $x_{16}$:'free', $x_{17}$:'0p...9p', $x_{18}$:'EURO, GBP, pound,$\mathscr{L}$, \$, €', $x_{19}$:'price'

(5) Other purpose (2): $x_{13}$:'sex', $x_{14}$:'girl'.

**Decomposition 2** As some features may have approximate meanings or roles, they could be combined together to be one feature, which can be implemented easily with the excel formula: 'if(or($x_1$,$x_2$,...$x_k$),1,0)'. For example, 'Award' and 'Prize' are combined to be one feature, 'award/prize', similarly, 'sex/girl', 'reply/send', 'cash/price','free/0p..9p' ','stop/click'. Thus, the number of features is reduced to 14 from original 20. In addition, the purpose of 'sex/girl' could be to attract the attention of the message receivers. Hence, it is placed to the subset of attraction. The new set of 14 features are decomposed to the following four subsets.

(1) Interjection (3): $x_0$:'urgent', $x_1$:congrate',$x_2$:'!'

(2) Attraction (4): $x_3$:'WIN/WON', $x_4$:'offer', $x_5$ Award/Prize', $x_6$:'sex/girl'

(3) Behaviour (4): $x_7$:'call', $x_8$:'Reply/send', $x_9$:'stop/click', $x_1$0:'text/txt'

(4) Benefit (3): $x_{11}$:'cash/price', $x_{12}$:'free/0p...9p', $x_{13}$:'EURO, GBP, pound,$\mathscr{L}$, \$, €'

**Decomposition 3** In this decomposition, subset one and subset two in Decomposition 2 are combined to be one subset, as the general purpose of these features could be to attract the attention of message receivers. Therefore, the 14 features are decomposed to three subsets.

(1) Attraction (7): $x_0$:'urgent', $x_1$:congrate',$x_2$:'!', $x_3$:'WIN/WON', $x_4$:'offer', $x_5$ Award/Prize', $x_6$:'sex/girl'

(2) Behaviour (4): $x_7$:'call', $x_8$:'Reply/send', $x_9$:'stop/click', $x_{10}$:'text/txt'

(3) Benefit (3): $x_{11}$:'cash/price', $x_{12}$:'free/0p...9p', $x_{13}$:'EURO, GBP, pound,£, $, €'

### D. Impact of each subset on spam detection

**Results of Decomposition 1** The 20 features have been divided into five categories: (1) interjection ($S_1$), (2) attraction ($S_2$), (3) behaviour ($S_3$), (4) benefit ($S_4$), (5) other purpose ($S_5$). The data with a subset of features are used to train and test an LDT. The outputs of the trained LDT are the appropriate measures of 'ham' and 'spam' for each tested message. If the message is more appropriate to be described with 'ham', it will be identified to 'ham', otherwise, 'spam'. If the estimated label is the same as the real label of the message sample, the identification is a true decision making. In terms of definitions, TPR, TNR, and $A$ are calculated. The sensitivity, specificity and general accuracy for each subset of features are observed, and shown in Fig. 3 (a).

From Fig. 3 (a), it can be seen that the sensitivities are much lower than the specificities for each subset of features, and sensitivities are increasing as the order from $S_1$ to $S_4$, but $S_5$ has the lowest sensitivity. The subset of benefit related features ($S_4$) is with the largest sensitivity. This demonstrates the capability of each subset in identifying positives or avoiding false negatives. $S_1$, $S_2$ and $S_5$ have similar specificities, and $S_3$ and $S_4$ have similar specificities, but which are smaller than that of $S_1$, $S_2$ and $S_5$. $S_4$ obtains the highest general accuracy.

**Results of Decomposition 2** Decomposition 2 made some adjustment of features based on Decomposition 1. The number of features is reduced to 14, and $S_5$ is combined to $S_2$. From Fig. 3 (b), it can be seen that the sensitivities and general accuracies are increasing as the order from $S_1$ to $S_4$, but $S_1$ and $S_2$ obtain higher specificities than that $S_3$ and $S_4$ do.

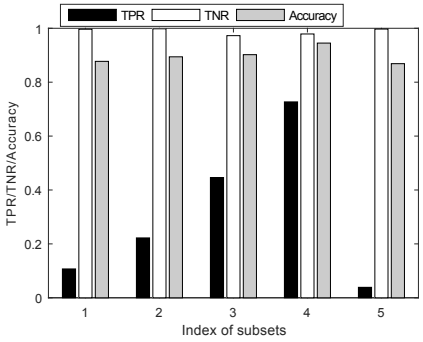
**Results of Decomposition 3** Fig. 3 (c) shows that the combination of $S_1$ and $S_2$ in Decomposition 2 improves the sensitivity, although the sensitivities still keep increasing in the order from $S_1$ to $S_3$ in Decomposition 3. It can be seen that the specificity of $S_1$ in Decomposition 3 does not drop as its sensitivity increases. Hence, it seems reasonable to combine $S_1$ and $S_2$ in Decomposition 2 to form the Decomposition 3.

### E. Integrated impact of LAHs

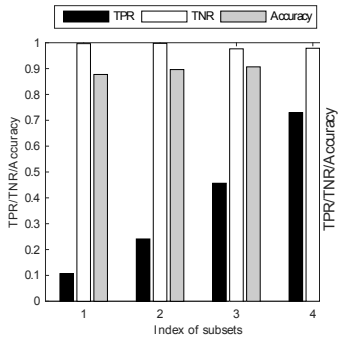
In this section, some LAHs embedded with LDTs are constructed and trained. Each LDT is fed with a subset of features. The construction of an LAH from bottom to top will be in terms of the sensitivity of identifying spams from low to high. Namely the LDTs are fed in turn with the order $S_5$, $S_1$, $S_2$, $S_3$, $S_4$.

**LAHs constructed with Decomposition 1** Five LDT-based LAHs (Figs. 5 - 9) are constructed and trained, and their performance, TPR TNR, Accuracy and the area under the ROC curve, are listed in Table II. An LDT is fed with a set $S$ of features is denoted as LDT($S$).

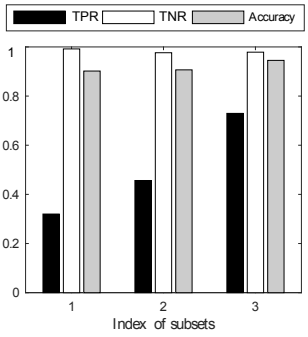
The first LAH with 5 subsets (denote as $LAH_1(5)$) is a cascade of LDTs, i.e. the output of the LDT in lower layer will be the input of the LDT in upper layer(see Fig. 5). In $LAH_2(5)$, shown in Fig. 6, LDT($S_5$) and LDT($S_1$) are at the bottom layer of the LAH. In $LAH_3(5)$, shown in Fig. 7, LDT($S_5$), LDT($S_1$), LDT($S_2$) are at the bottom layer of the LAH. In $LAH_4(5)$,




(a) Decomposition 1




(b) Decomposition 2




(c) Decomposition 3

Fig. 3. Sensitivity, Specificity, and Accuracy for Decompositions 1, 2 & 3

shown in Fig. 8, LDT($S_5$), LDT($S_1$), LDT($S_2$) and LDT($S_3$) are at the bottom layer of the LAH. In $LAH_5(5)$, shown in Fig. 9, LDT($S_5$), LDT($S_1$), LDT($S_2$), LDT($S_3$), LDT($S_4$) are all at the bottom layer of the LAH.

Table II lists all sensitivities, specificities, accuracies and the areas under ROC curves for the five LAHs. It can be seen that $LAH_1(5)$ - $LAH_4(5)$ have similar performance. $LAH_5(5)$ has different performance to that the previous four LAHs have, its sensitivity drops, and specificity increases. But the accuracy of $LAH_5(5)$ is close to that of $LAH_2(5)$, and $LAH_5(5)$ obtained the largest area under the ROC curve. Namely, when the LDT($S_4$) is placed at the same layer with other four LDTs, the features in other subsets could affect the performance, so that the sensitivity of $LAH_5(5)$ drops, not keeping the level of sensitivity obtained by LDT($S_4$) at different layers.

TABLE II. TPR, TNR AND ACCURACY OF LINGUISTIC ATTRIBUTE HIERARCHIES CONSTRUCTED WITH DECOMPOSITION 1

| LAHs | TPR | TNR | A | $A_{ROC}$ |
|---|---|---|---|---|
| $LAH_1(5)$, Fig. 5 | 0.717537 | 0.987156 | 0.951023 | 0.945513 |
| $LAH_2(5)$, Fig. 6 | 0.721553 | 0.987570 | 0.951920 | 0.947119 |
| $LAH_3(5)$, Fig. 7 | 0.710843 | 0.985913 | 0.949049 | 0.948952 |
| $LAH_4(5)$, Fig. 8 | 0.732262 | 0.978869 | 0.945820 | 0.948518 |
| $LAH_5(5)$, Fig. 9 | 0.650602 | 0.998135 | 0.951561 | 0.958341 |

**LAHs constructed with Decomposition 2** Similarly, four LAHs are constructed, trained and tested. In $LAH_1(4)$, shown in Fig. 10, LDT($S_1$) - LDT($S_4$) are cascaded at the different layers of the LAH. In $LAH_2(4)$, shown in Fig. 11, LDT($S_1$)

and LDT($S_2$) are at the bottom layer of the LAH. In $LAH_3(4)$, shown in Fig. 12, LDT($S_1$), LDT($S_2$), LDT($S_3$) are at the bottom layer of the LAH. In $LAH_4(4)$, shown in Fig. 13, LDT($S_1$), LDT($S_2$), LDT($S_3$), LDT($S_4$) are all at the bottom layer of the LAH.

Their performance in TPR, TNR, Accuracy and the area under ROC curve are listed in Table III. The results are similar to that of Decomposition 1. $LAH_4(4)$ has the best performance in specificity, accuracy and the area under ROC, but its sensitivity is worse than that of other LAHs.

TABLE III. TPR, TNR, ACCURACY, $A_{ROC}$ OF LINGUISTIC ATTRIBUTE HIERARCHIES CONSTRUCTED WITH DECOMPOSITION 2

| LAHs | TPR | TNR | A | $A_{ROC}$ |
|---|---|---|---|---|
| $LAH_1(4)$, Fig. 10 | 0.741633 | 0.978455 | 0.946717 | 0.950168 |
| $LAH_2(4)$, Fig. 11 | 0.741633 | 0.978455 | 0.946717 | 0.956266 |
| $LAH_3(4)$, Fig. 12 | 0.741633 | 0.978455 | 0.946717 | 0.958916 |
| $LAH_4(4)$, Fig. 13 | 0.625167 | 0.998550 | 0.948511 | 0.956282 |

**LAHs constructed with Decomposition 3** Again, with the same strategy, three LAHs are constructed, trained and tested. In $LAH_1(3)$, shown in Fig. 14, LDT($S_1$) - LDT($S_3$) are cascaded at the different layers of the LAH. In $LAH_2(3)$, shown in Fig. 15, LDT($S_1$) and LDT($S_2$) are at the bottom layer of the LAH. In $LAH_3(3)$, shown in Fig. 16, LDT($S_1$), LDT($S_2$) and LDT($S_3$) are all at the bottom layer of the LAH.

The performance in TPR, TNR, Accuracy and the area under ROC for the three LAHs are listed in Table IV. Similar to the previous two decompositions, the specificity, accuracy and the area under ROC are increasing (non-decreasing) as the place of LDT($S_4$) drops from top to bottom. But the sensitivity of $LAH_2(3)$ is larger than that of $LAH_1(3)$ while $LAH_2(3)$ keeps the same specificity as $LAH_1(3)$. However, $LAH_3(3)$ is with the dropped sensitivity again as the LAHs for previous two decompositions.

TABLE IV. TPR, TNR AND ACCURACY OF LINGUISTIC ATTRIBUTE HIERARCHIES CONSTRUCTED WITH DECOMPOSITION 3

| LAHs | TPR | TNR | A | $A_{ROC}$ |
|---|---|---|---|---|
| $LAH_1(3)$, Fig. 14 | 0.736278 | 0.978455 | 0.945999 | 0.948868 |
| $LAH_2(3)$, Fig. 15 | 0.742972 | 0.978455 | 0.946896 | 0.955243 |
| $LAH_3(3)$, Fig. 16 | 0.654618 | 0.996271 | 0.950484 | 0.960311 |

### *F. Comparison of best solutions for different decompositions*

**(1) Best performance in accuracy** Table V lists the performance of the solutions that obtained the best accuracy in all LAHs, constructed with different decompositions. For

TABLE V. TPR, TNR, ACCURACY, AREA UNDER ROC OF LAHS WITH THE HIGHEST ACCURACY CONSTRUCTED WITH DIFFERENT DECOMPOSITIONS

| LAHs | TPR | TNR | A | $A_{ROC}$ |
|---|---|---|---|---|
| $LAH_2(5)$, Fig. 6 | 0.721553 | 0.987570 | 0.951920 | 0.947119 |
| $LAH_4(4)$, Fig. 13 | 0.625167 | 0.998550 | 0.948511 | 0.956282 |
| $LAH_3(3)$, Fig. 16 | 0.654618 | 0.996271 | 0.950484 | 0.960311 |

Decompositions 2 and 3, the LAHs with all LDTs at the bottom layers of the LAHs obtained best accuracy, but both have lowest sensitivities. For Decomposition 1, the best solution is not the LAH with all LDTs at the bottom layers of the LAH, but the second LAH, in which LDT($S_1$) and LDT($S_2$) are at the bottom layer of the LAH. This may indicate $S_1$ and $S_2$ together could improve the accuracy. Regarding sensitivity, $LAH_2(5)$ is the best. Regarding the area under ROC, $LAH_3(3)$ is the best.

**(2) Best ROC curves for different decompositions** Table VI lists the performance of LAHs with the best area under the ROC curve for the three decompositions. Two single LDTs were trained and tested with the two full sets (20 and 14) of features, and their performance is listed in the table as well. Fig. 4 shows the ROC curves for the five solutions. Regarding the area under ROC curve, $LAH_3(3)$ is the best in the all listed solutions; regarding specificity, $LAH_5(5)$ is the best; and regarding sensitivity and accuracy, LDT(14) is the best. It is surprising that the two single LDTs have worse performance in the area under ROC curves than the listed LAHs, as this is different to the convention.

TABLE VI. TPR, TNR AND ACCURACY OF LINGUISTIC ATTRIBUTE HIERARCHIES WITH HIGHEST $A_{ROC}$

| LAHs | TPR | TNR | A | $A_{ROC}$ |
|---|---|---|---|---|
| $LAH_5(5)$, Fig. 9 | 0.650602 | 0.998135 | 0.951561 | 0.958341 |
| $LAH_3(4)$, Fig. 12 | 0.741633 | 0.978455 | 0.946717 | 0.958916 |
| $LAH_3(3)$, Fig. 16 | 0.654618 | 0.996271 | 0.950484 | 0.960311 |
| LDT($S_{20}$) | 0.757697 | 0.991092 | 0.959813 | 0.881349 |
| LDT($S_{14}$) | 0.769746 | 0.990470 | 0.960890 | 0.912419 |

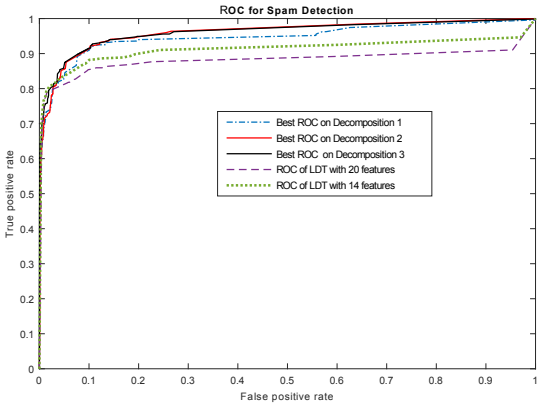



Fig. 4. Best ROC curves of LAHs for each decomposition and the two single LDTs

### *G. Comparison of branch numbers of different solutions for different decompositions*

If we do not set the threshold of probability during the training process of an LDT, fed with $n$ features, each branch in the single LDT will have $n$ edges. If we use each focal element to represent an edge in an LDT, and each variable (feature) could be described with $\tau$ focal elements, then each non-leaf node in an LDT will have $\tau$ child nodes. Namely, the single LDT is a $\tau$-branch tree with $\tau^n$ leaves (branches). In the implementation, the learning threshold is set to the maximum 1. Namely, if the probability of positive or negative gets 1, the node will be a leave (i.e. no further learning process). Therefore, real number of branches of a single LDT is less than the maximum. Assume $\rho_i$ is the number of input attributes for $LDT_i$ in an LAH. An LDT will be a $\tau$-branch tree with $\tau^{\rho_i}$ branches. The number of total branches in a hierarchy:

$$\beta(H) = \sum_i (\beta_i) = \sum_i (\tau^{\rho_i}). \tag{8}$$

TABLE VII. BRANCH NUMBERS OF ALL SOLUTIONS, WHERE $\beta$ DENOTES THE NUMBER OF BRANCHES

| Decomp. 1 | | Decomp. 2 | | Decomp. 3 | | Single LDT | |
|---|---|---|---|---|---|---|---|
| Figs | $\beta$ | Figs | $\beta$ | Figs | $\beta$ | $n$ | $\beta$ |
| Fig. 5 | 237 | Fig. 10 | 138 | Fig. 14 | 151 | 20 | 697 |
| Fig. 6 | 269 | Fig. 11 | 146 | Fig. 15 | 157 | 14 | 421 |
| Fig. 7 | 471 | Fig. 12 | 150 | Fig. 16 | 106 | | |
| Fig. 8 | 212 | Fig. 13 | 105 | | | | |
| Fig. 9 | 202 | | | | | | |

Table VII lists the branch numbers of the single LDTs and all solutions in Figs. 5 - 16 in Appendix.

From Table VII, it can be seen that the branch numbers of all LAHs in Figs. 5 - 9 for Decomposition 1 are less than that (697 branches) of the single LDT, trained with the 20 attribute data set, and the branch numbers of all LAHs in Figs. 10-16 for Decompositions 2 and 3 are less than that (421 branches) of the single LDT, trained with the 14 attribute data set. Moreover, due to the set of features are decomposed to subsets of features, the number of input features for each LDT in an LAH is less than the number $n$ of total features. Namely, the length of each branch is short than that of branches in the single LDT. This indicates the computing complexity is reduced very much.

## IV. CONCLUSIONS

In this paper, we proposed another form of deep learning, a linguistic attribute hierarchy, embedded with linguistic decision trees, for spam detection. A case study was carried out on the SMS message database from the UCI machine learning repository. It has been shown that the decomposition of features plays an important role in the construction of the linguistic attribute hierarchy, which directly affects the performance of decision making by the constructed linguistic attribute hierarchy. In the experiments, three decompositions of features (attributes) were investigated, and the LAHs were constructed based on strategies for the three decompositions semantically. The performance of LAHs for different decompositions were examined in term of sensitivity, specificity, accuracy and ROC curve. According to the experimental results, the LDT with the highest sensitivity should stay in the top layer, different to the layers where other LDTs with lower sensitivity in order to obtain a high sensitivity of the LAH, but in order to obtain large area under ROC, all LDTs should be placed in the bottom layer of the LAH. However, without decomposition, a single LDT trained by the full set of features cannot obtain a larger area under ROC than an LAH could have. The experimental results show that the features related to the benefits and finance have important impact on the sensitivity of identifying spam messages. This observation matches the convention of human knowledge. The process of experiments has demonstrated the use of a hierarchy of linguistic decision tree approach for deep learning of feature impact on spam detection. Hence, an LAH, embedded with LDTs, provides a transparent approach to in-depth analysing feature impact to the spam detection. This approach can not only improve the performance of spam detection when the semantic attributes are constructed to a proper hierarchy, but also efficiently tackle 'curse of dimensionality' in spam detection with massive attributes. The automatic knowledge based decomposition of features and the optimisation of linguistic attribute hierarchies and high performance spam detectors will be the future work.

## REFERENCES


[1] N. Spirin and J. Han. Survey on web spam detection: Principles and algorithms. ACM SIGKDD Explorations Newsletter archive,ACM New York, NY, USA, 13(2), 12 2011.

[2] T. A. Almeida, J. M. G. Hidalgo, and A. Yamakami. Contributions to the study of sms spam filtering:new collection and results. In DocEng'11, Mountain View, California, USA, 19-22 September 2011.

[3] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. Techincal report, Stanford University, 1998.

[4] H. Garcia-Molina Z. Gyöngyi and J. Pedersen. Combating web spam with trustrank. In 30th International Conference on Very Large Data Bases (VLDB), pages 576–587, Morgan Kaufmann, August 2004. Morgan Kaufmann.

[5] V. Krishnan and R. Raj. Web spam detection with anti-trust rank. In AIRWEB, pages 37–40, 2006.

[6] A. A. Benczúr, K. Csalogány, T. Sarlás, and M. Uher. Spamrank - fully automatic link spam detection work in progress. In The first international workshop on adversarial information retrieval on the web (AIRWeb'05), Chiba, Japan, 2005.

[7] M. Richardson, A. Prakash, and E. Brill. Beyond pagerank: Machine learning for static ranking. In 15th International Conference on World Wide Web (WWW), pages 707–715, NY, USA: ACM Press, May 2006.

[8] R. M. Silva, A. Yamakami, and T. A. Almeida. An analysis of machine learning methods for spam host detection. In 2012 11th International Conference on Machine Learning and Applications (ICMLA), volume 2 of 10.1109/ICMLA.2012.161, Boca Raton, FL, Dec 2012.

[9] B. Wang, A. Zubiaga, M.Liakata, and R. Procter. Making the most of tweet-inherent features for social spam detection on twitter. In Proceedings of the 5th Workshop on Making Sense of Microposts( Microposts2015) @WWW2015, volume CEUR 1395, pages 10–16, Florence, Italy, 18 May 2015.

[10] M. Crawford, T. M. Khoshgoftaar, J. D. Prusa, A. N. Richter, and H. A. Najada. Survey of review spam detection using machine learning techniques. Journal of Big Data, 2(23):Online open assess, Dec 2015.

[11] H. Shirani-Mehr. SMS spam detection using machine learning approach, 2013. http://cs229.stanford.edu/proj2013/ShiraniMehr-SMSSpamDetectionUsingMachineLearningApproach.pdf. accessed on 26 Sept 2016.

[12] J. Verma and S. Dhawan. Detection of spam in social networks using clustered k-nearest neighbour. International Journal of Advanced Research in Computer Science and Software Engineering, 5(3):1120–1123, Mar 2015.

[13] M. Sahami, S. Dumais, D. Heckerman, and E. Horvitz. A bayesian approach to filtering junk email. In AAAI-98 Workshop on Learning for Text Categorization, Madison, Wisconsin, USA, 1998.

[14] GFI Software. Why bayesian filtering is the most effective anti-spam technology. GFI White Paper, 2011.

[15] K. Tretyakov. Machine learning techniques in spam filtering. In Data Mining Problem-oriented Seminar, MTAT.03.177, pages 60–79, May 2004.

[16] J. Alqatawna, H. Faris, K. Jaradat, M. Al-Zewairi, and O. Adwan. Improving knowledge based spam detection methods: The effect of malicious related features in imbalance data distribution. International Journal of Communications, Network and System Sciences, 8:118–129, 2015.

[17] H. He, A. Tiwari, J. Mehnen, T. Watson, C. Maple, Y. Jin, B. Gabrys, Incremental Information Gain Analysis of Input Attribute Impact on RBF-Kernel SVM Spam Detection, WCCI2016, Vancouver, Canada, 24-29 July, 2016.

[18] H. He and J. Lawry, Optimal Cascaded Hierarchies of Linguistics Decision Trees for Decision Making, in Proc. of The IAENG International Conference on Artificial Intelligence and Applications (ICAIA'09), Hong Kong, 18-20 March, 2009, pp. 1-6.

[19] H. He and J. Lawry, Linguistic Attribute Hierarchy and Its Optimisation for Classification Problems, Soft Computing, 18(10), pp. 1967-1984, Oct. 2014. DOI: 10.1007/s00500-013-1179-3.

[20] H. He, Z. Zhu, A. Tiwari and Andrew Mills, A Cascade of Linguistic CMAC Neural Networks for Decision Making, in Proceedings of

the International Joint Conference on Neural Networks (IJCNN2015), Killarney, Ireland, 12-17 July, 2015.

[21] J. Lawry, *Modeling and Reasoning with Vague Concepts*, (Kacprzyk, J. Ed.), Springer, 2006.

[22] Jeffrey RC (1990) The Logic of Decision, The University of Chicargo Press, New York.

[23] Qin Z (2005) ROC analysis for predictions made by probabilistic classifiers, in Proc. of the International Conference on Machine Learning and Cybernetics, 5:3119- 3124.

[24] Quinlan JR (1986) Induction of Decision Trees, Machine Learning, 1:81-106. Kluwer Academic Publishers, Boston.

[25] Lichman, M. (2013). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.

## APPENDIX

### A. LAHs based on Decomposition 1 (Figures 5-9)

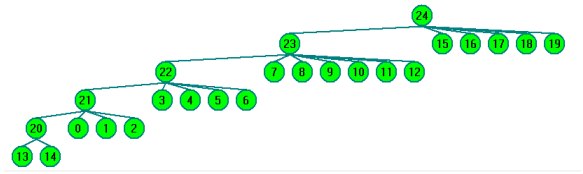


Fig. 5. Cascade of five LDTs in the LAH

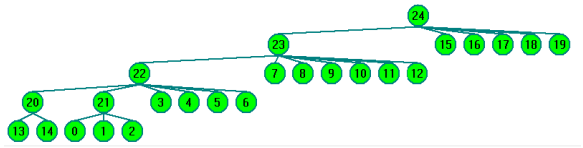


Fig. 6. LDT1 and LDT2 are at the same level in the LAH

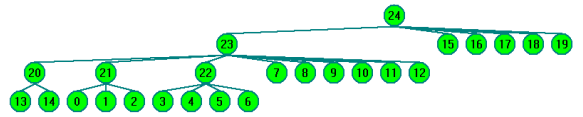


Fig. 7. LDT1, LDT2 and LDT3 are at the same level in the LAH

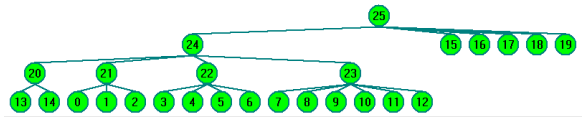


Fig. 8. LD1 - LD4 are at the same level in the LAH

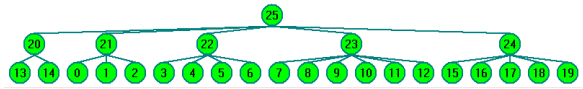


Fig. 9. All five LDTs are at the same level in the LAH

### B. LAHs based on Decomposition 2 (Figures 10-13)

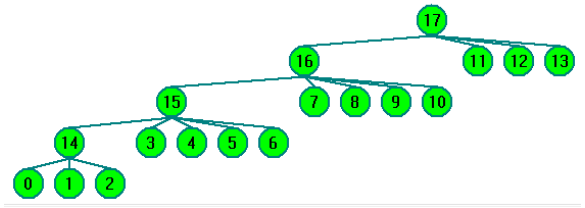


Fig. 10. A cascade of four LDTs in the LAH

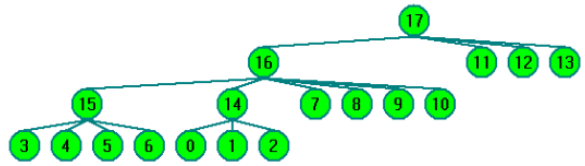


Fig. 11. LDT1 and LDT2 are at the same level in the LAH

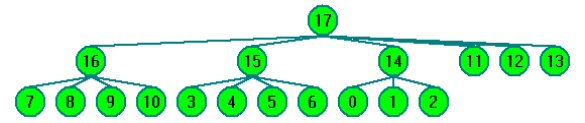


Fig. 12. LD1 - LD3 are the same level in the LAH

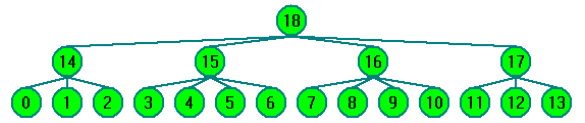


Fig. 13. All LDTs are at the same level in the LAH

### C. LAHs based on Decomposition 3 (Figures 14-16)

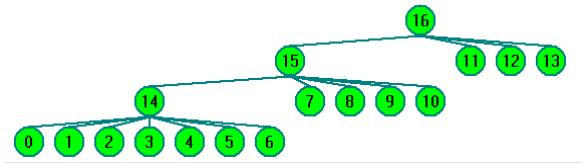


Fig. 14. A cascade of LDTs in the LAH

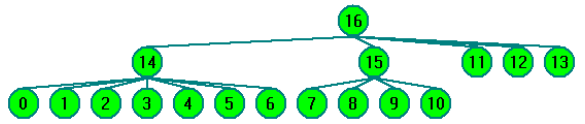


Fig. 15. LDT1 and LDT2 are at the same level in the LAH

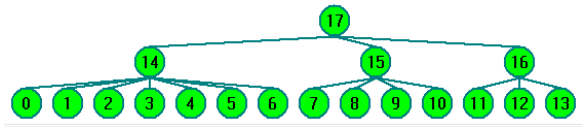


Fig. 16. All LDTs are at the same level in the LAH