PROJECT REPORT

CZ4041: MACHINE LEARNING

RESEARCH TOPIC: SEMI-SUPERVISED LEARNING

Instructor: Asst Prof. Pan, Sinno Jialin

CHAN YI HAO	U1520808B
MICHELLE LIM SHI HUI	U1522038H
SOH JUN JIE	U1521123B
TONI MIHARJA	U1521012E

NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

ABSTRACT

Semi-Supervised Learning (SSL) is a learning paradigm concerned with the design of models in the presence of both labeled and unlabeled data. SSL has gained significant attention due to its ability to perform its own labelling of unlabeled training dataset and learn from its experience. Focusing on SSL, the purpose of this paper can be divided into the following 2 key objectives:

1. Explore state-of-the-art SSL methods so as to expand the current SSL taxonomy
2. Propose an extension of the application of DuAL Learning-based sAfe Semi-supervised learning (DALLAS) to create a SSL model that perform well when given both labeled and unlabeled data

The motivation behind the first objective is the emergence of several state-of-the-art methods in SSL. This paper aims to study newly proposed SSL methods more closely and expand the existing SSL method taxonomy proposed by Triguero et al. (2013).

The second objective revolves around proposing a new SSL method revolving around DALLAS, a method employing dual learning to estimate the safety or risk of the unlabeled instances to improve SSL performance. This paper proposes an extension of its application, utilising risk-based classification of unlabeled data, Regularised Least Squares (RLS) prediction model and traditional supervised learning (SL) algorithm. Analysis of the performance of the proposed model is also conducted, showing that it outperforms traditional SSL methods, exhibiting performance close enough to fully supervised methods.

GROUP MEMBER ROLES

Name	Role and Responsibilities
Chan Yi Hao	Project planning, introduction, overview of SSL, literature review
Michelle Lim Shi Hui	Literature review of existing and state-of-the-art SSL algorithms
Soh Jun Jie	Literature review and implementation of state-of-the-art methods Code implementation of DALLAS
Toni Miharja	Overall report structure, abstract and introduction Implement and analyse performance of proposed method using different algorithms

TABLE OF CONTENTS

1. INTRODUCTION

Semi-Supervised Learning (SSL) is a learning paradigm concerned with the design of models in the presence of both labelled and unlabelled data (Triguero et al., 2013). In such situations, supervised learning can be performed on the labelled dataset, while unsupervised learning can be performed on the unlabelled dataset. SSL aims to (i) augment the performance of supervised learning by using information from the unused unlabelled data and (ii) augment the performance of unsupervised learning by using information from the unused labelled data.

The main use case for SSL is in the context where unlabeled data is abundant but labels are expensive to get, causing the amount of unlabelled data available to be significantly more than labelled data. Such a problem is prevalent in the medical (Li & Zhou, 2007), (Cheplygina et al, 2018) and language processing domain (Yu et al, 2011), to name a few. Research in SSL is also driven by the motivation of being able to model human learning (Zhu et al., 2010), which is believed to be made up more general observations than direct instructions (Zaki & Nososky, 2007). This closely parallels SSL as labelled data acts like a parent or a teacher teaching a child, while learning from unlabelled data is akin to a child figuring out the structure of the world via observation.

SSL is currently seeing a resurgence in popularity. Google used graph-based SSL approaches in Gmail's Smart Reply feature when they did not have a labelled dataset for semantic intent and it was not feasible to create a comprehensive one (Kannan et al., 2016). Most recently, SSL has been successfully applied in Amazon Alexa where it reduced the amount of labelled data needed to achieve the same accuracy improvement by 40 times (Bezos, 2018).

With its popularity, much research has been done in classifying the various SSL methods that have been previously implemented. There has been several efforts to compile the progress in SSL research (Chapelle et al, 2006), (Zhu, 2008), (Zhu et al., 2009) and forming taxonomies (Triguero et al., 2013). However, since then, many state-of-the-art methods have been proposed. In particular, Deep SSL - SSL applied in deep learning models - has emerged as a new subfield. Ladder Networks (Rasmus et al, 2015) and Pseudo Labelling (Lee, 2013) are examples of such new approaches which will be expounded on in the later sections of this paper. To evaluate the recent developments in this field, it is imperative to look at whether these novel approaches can be classified using the previous taxonomy or whether there is a need to expand the existing taxonomy. As such, exploring these state-of-the-art methods will be one area of focus of this paper.

One other key interest in SSL is Safe SSL. It entails the idea of having the ability to differentiate "risky" unlabelled data sets from the rest of the unlabelled set to improve the performance of semi-supervised learning (Gan et al., 2018). This is because a certain unlabeled data may degenerate the performance of the model, making it "risky" to use them in generating the SSL model. This is proven in several researches where empirical and theoretical analysis are conducted (Gan et al, 2013), showing the negative effect of unlabeled instances, ultimately limiting the application scope of SSL due to the inaccuracies involved (Yang et al., 2011). One approach to tackle this problem is proposed by Gan et al. where DuAL Learning-based sAfe Semi-supervised learning (DALLAS) is proposed, employing dual learning to estimate the safety or risk of the unlabeled instances. The method has been shown to improve the performance of semi-supervised classification algorithms. With the ability to differentiate “risky” data from the “safe” data, DALLAS can potentially be applied and be extended to improve the performance of any model that utilises unlabeled data. With its versatility, this recently proposed method will be studied in this paper. Moreover, an extension of its application to create a SSL model that perform well when given both labeled and unlabeled data is proposed.

In summary, the purpose of this paper can be divided into the following 2 key objectives:

1. Explore state-of-the-art SSL methods so as to expand the current SSL taxonomy
2. Propose an extension of the application of DALLAS to create a SSL model that perform well when given both labeled and unlabeled data

The remaining sections of the paper are organised as follows:

- Section 2 gives a big picture overview of where SSL stands in machine learning and also defines the scope of our paper.
- Section 3 provides a literature review of existing SSL approaches and updates the taxonomy provided by Triguero et al.
- Section 4 summarizes the methodology used in the DALLAS paper to calculate the risk for each dataset instance.
- Section 5 explains our proposed extension to DALLAS.
- Section 6 shows the experimental results collected and the limitations of our study.
- Section 7 concludes the paper by summarising our findings and suggesting possible future areas of work.
- Appendix contains detailed information about the dataset and algorithms used for our experiments.

2. OVERVIEW OF SSL

2.1 Scope of Discussion

SSL can be seen to be halfway between unsupervised learning and supervised learning as mentioned by Chapelle et al. A distinction is commonly made between inductive and transductive SSL. Concretely, using the notations from (Zhu eoml, 2007), we define an instance of our input from the dataset as x having a label y. Our dataset is then thus made up of labelled data $\{(x_i, y_i)\}_{i=1}^{l}$and unlabelled data $\{x_i)_{i=l}^{i=l+u}, l << u$. In inductive SSL, our aim is to learn a function $f : X \rightarrow Y$, where $x \in X, y \in Y$. On the other hand, transductive SSL does not require such generalisation: it is only concerned with the label predictions of $\{x_i)_{i=l}^{i=l+u}$. We will focus on inductive SSL.

SSL can also be seen from two main perspectives: (i) making unsupervised learning more informed ("semi-supervised clustering") or enhancing supervised learning performance on the partially labelled training set ("semi-supervised classification" or "semi-supervised regression"). As a comprehensive review of all of them would be infeasible, we will focus on Semi-Supervised Classification (SSC).

2.2 Assumptions in SSL

SSL requires some assumptions about the dataset to be met for it to work well, such that the information gained from the unlabelled dataset will be able to help improve the performance of our model f. Chapelle et al. suggested the following 3 assumptions:

1. Smoothness assumption: If two points $x_1$and $x_2$in a high-density region are close, then so should be the corresponding outputs $y_1$and y_2.
2. Cluster assumption: If points are in the same cluster, they are likely to be of the same class.
3. Manifold assumption: The high-dimensional data lie roughly on a low-dimensional manifold.

When assumptions are not met, the performance of SSL could be even worse than that of supervised learning (Zhu, 2007).

2.3 Related Work

With the ability to perform its own labelling of unlabelled training dataset and learn from its experience, SSL has attracted much attention. However, SSL is one of several approaches that uses unlabelled data to improve the performance of current models. Other approaches includes active learning, transfer learning and dual learning (Gan et al., 2018). Given the similarities that these approaches have, ideas from those areas of research could be applied to SSL as well. In the later sections of this paper, the focus will be on an approach that is inspired from dual learning.

3. LITERATURE REVIEW

Over the years, much research has been extensively conducted into developing algorithms to perform Semi-Supervised Classification (SSC). Triguero et al. (2013) has previously done a comprehensive overview of Self-Labelling techniques and classified them with a taxonomy. Xu et al. (2013) has also done a similar survey on a multi-view methods. However, there has not been any taxonomy created to classify all the SSC algorithms.

In this section, we aim to extend these proposed taxonomies to include newer additions to the Self-Labelling family, as well as to consider several other approaches. There are currently 6 main families of SSC algorithms:

1. Self-Labelling Techniques
2. Support Vector Machine (SVM) Models
3. Graph-based Models
4. Deep Generative Models (Not discussed in this paper)
5. Neural Network Models
6. Safe SSL

The final taxonomy can be found in Section 3.6, Figure 1.

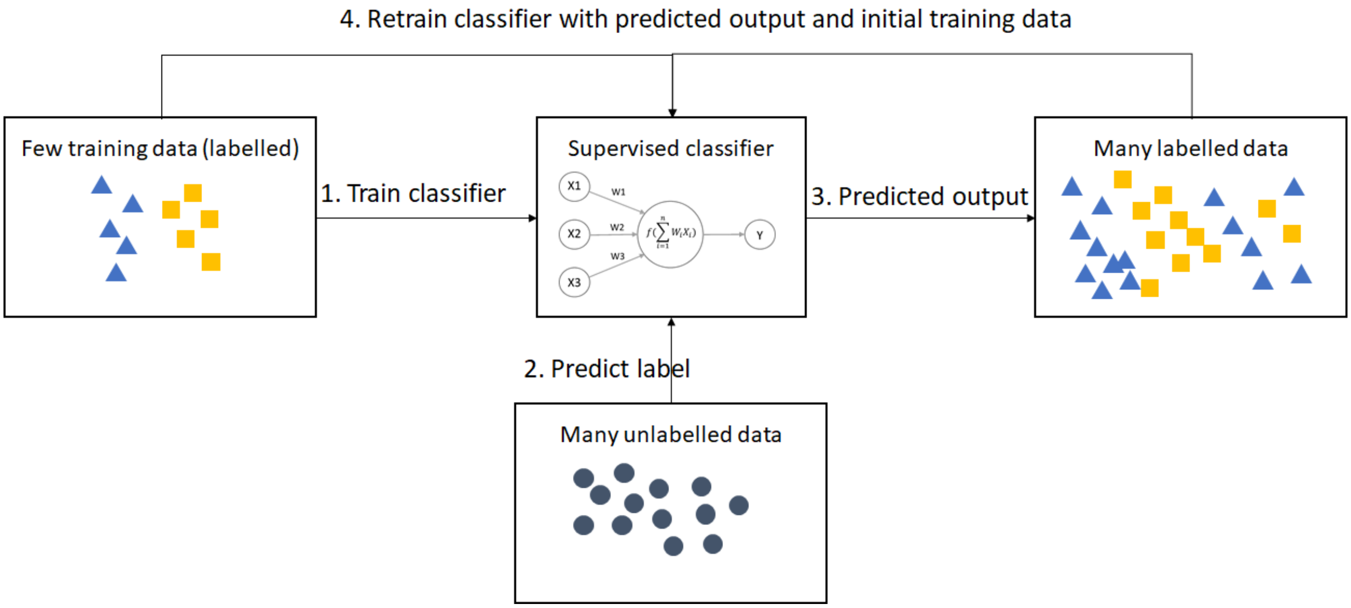
3.1 Self-Labelling Techniques

Self-Labelling involves one or more learner(s) teaching itself or each other, enlarging the training set with its most confident labels, in an iterative process.

Self-Labelling techniques are often popular because they do not require making any distribution assumptions. However, since poor classifications may be reinforced, these methods can also be prone to error, resulting in a final model with poor accuracy.

We borrow the taxonomy proposed by Triguero et al. (2013) to classify the work historically performed in this field into 4 broad categories.

3.1.1 Single Learner, Single View, Single Algorithm (Self-Training)



Self-training is a commonly used technique for semi-supervised classification, pioneered by D. Yarowsky (1995). In self-training, a classifier is firstly trained with a small number of labeled training example. This classifier is then applied to the entire sample set, and its most confident predictions will be labelled and added back into the training set for retraining. This cycle continues until a stable residual set is formed to form a final classifier, which it taught itself based on its own past predictions.

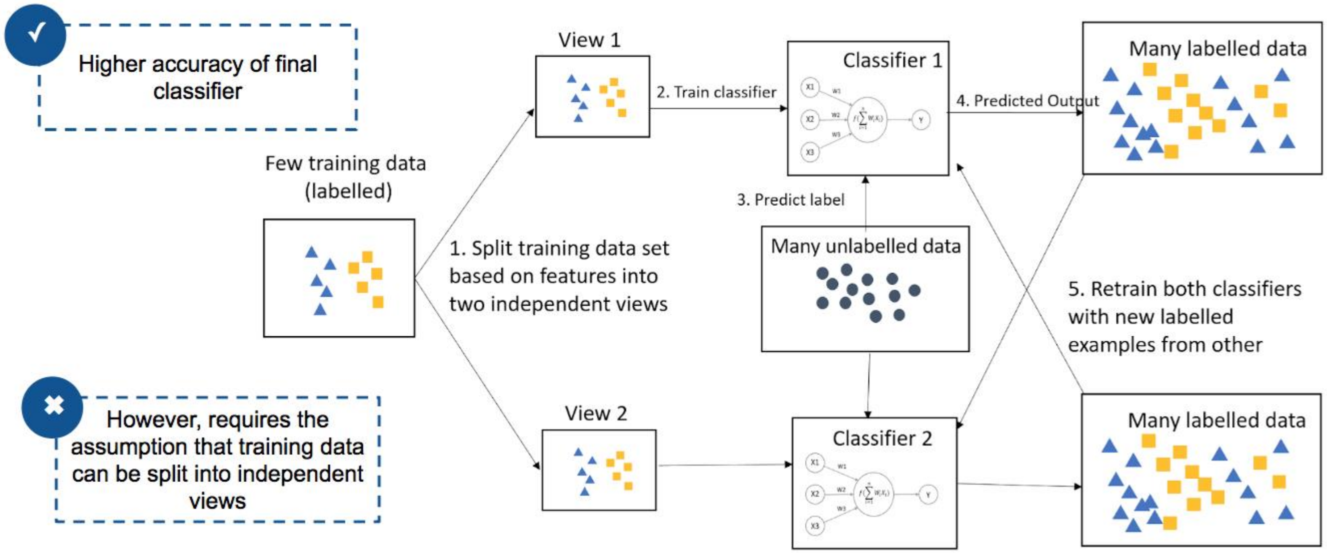
However, since the labeled set is usually insufficient for learning, misclassifying a certain amount of unlabeled data is unavoidable (M. Li & Z.H. Zhou, 2005). Such noisy examples continue to affect the learner due to the iterative nature of the process, results in a degrading of the generalization ability of the final classifier. Li & Zhou (2005) therefore proposed Self-Training with Editing (SETRED), which incorporated a data editing technique to actively identify possibly mislabeled examples from the set of self-labeled examples at every iteration, the model is more robust to noisy examples.

Other variations of the self-training family include Self-training Nearest Neighbor Rule using Cut Edges (SNNRCE), which reduced classification errors in early iterations by constructing a relative neighbourhood graph and using the cut edge weight as the basis for classification (Y. Wang et al, 2010); as well as the Aggregation Pheromone Density based Semi-Supervised Classification (APSSC), where the property of Aggregation Pheromone (AP) found in ants was used to add self-labelled examples in a batch approach. (A. Halder et al., 2013)

Fazakis et al. (2015) have also recently proposed a new method under the self-training family, which combines the self-training scheme with the Logistic Model Tree (LMT) algorithm, whereby a decision tree with piecewise linear regression model is used to generate the estimates for the class membership probabilities.

3.1.2 Multiple Learners (Co-training) methods

3.1.2.1 Traditional Co-training (Single algorithm, multiple views)


✓
Higher accuracy of final classifier
View 1
2. Train classifier
Classifier 1
X1
X2
X3
W1
W2
W3
Y
4. Predicted Output
Many labelled data
Few training data (labelled)
3. Predict label
1. Split training data set based on features into two independent views
Many unlabelled data
5. Retrain both classifiers with new labelled examples from other
✖
However, requires the assumption that training data can be split into independent views
View 2
Classifier 2
Many labelled data

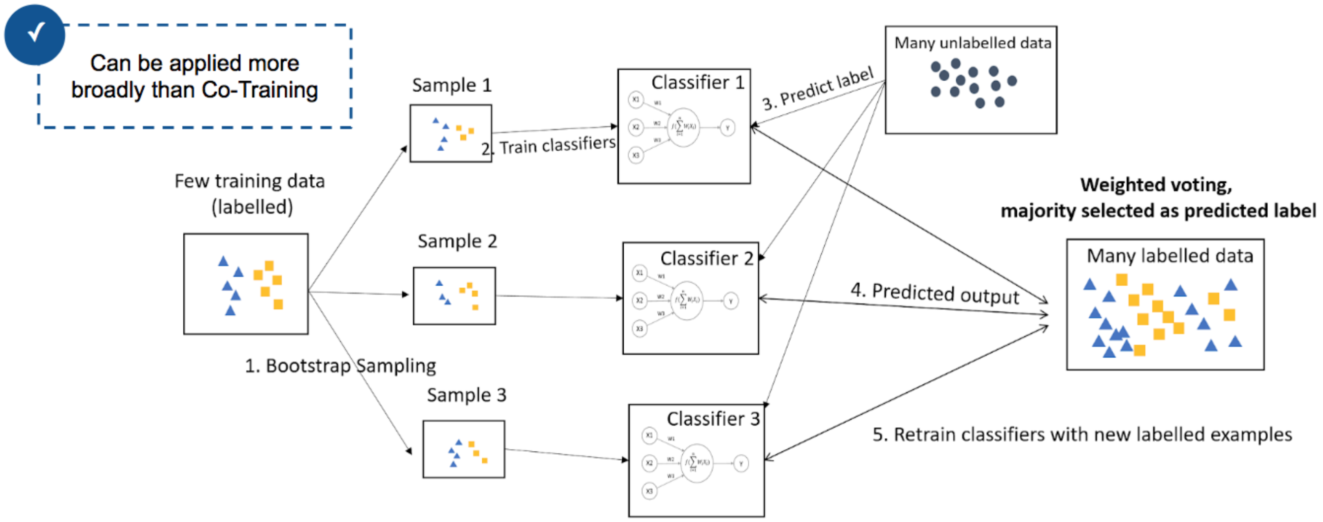
Co-training is another commonly employed semi-supervised classification method. It assumes that the training set can be split into two independent and redundant views, and that either view of the example would be sufficient for learning. (A. Blum & T. Mitchell., 1998) One classifier will be trained on each view, and each classifier's predicted self-labelled examples will be used to enlarge the training set of the other as well as its own. In other words, the two classifiers teach each other based on the different patterns in the data they recognize. Dasgupta et al.(2010) has proven that when this requirement is met, co-training of these two classifiers could make fewer generalization errors by maximizing their agreement over the unlabeled data.

J. Wang et al. (2008) then extends the co-training algorithm with two views to a multi-view situation with the Random Subspace Co-Training (RASCO) algorithm, where random subspaces are chosen and a classifier is trained on each subspace. Relevant RASCO (Rel-RASCO) then improves the accuracy of RASCO by using probability to pick only features that are proportional to their relevances measured by the mutual information between features and class labels Y. (Yaslan et al, 2010).

Sun & Jin (2011) then introduced the integration of canonical correlation analysis (CCA), to improve the evaluation of self-labelled instances. Predicted labels will be checked against a closely correlated representation of the original data, ensuring consistency before it can be used to enlarge the original training set.

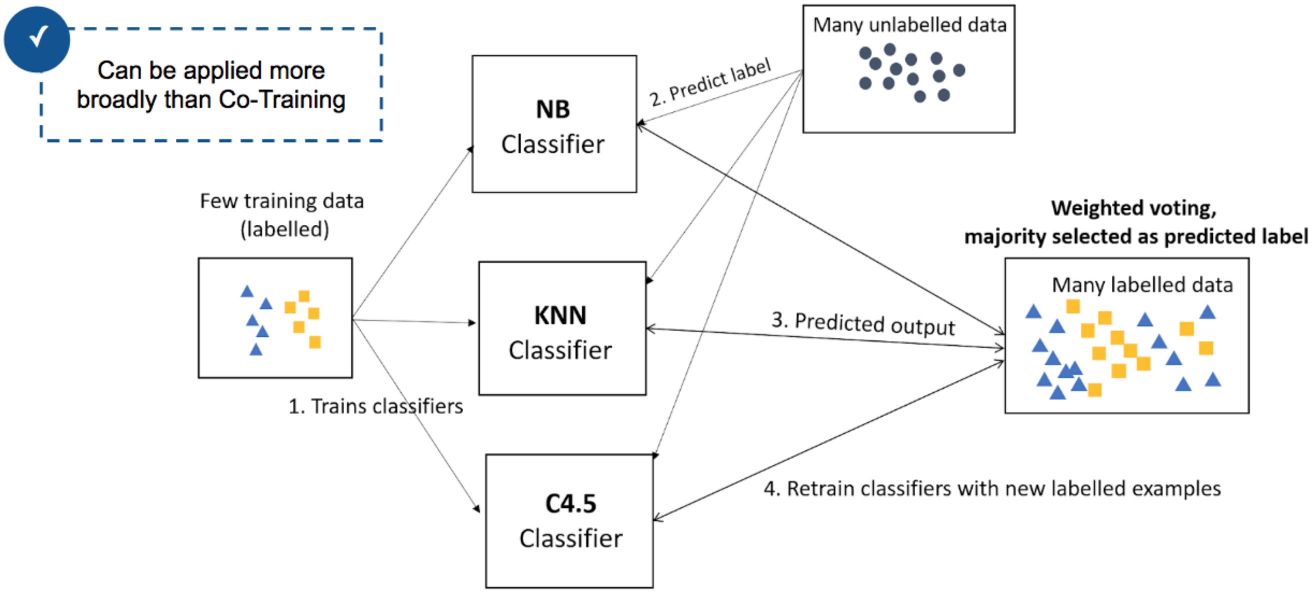
Yet another approach explored was to combine multiple algorithms into a single hybrid method. Jiang et al. (2013) combined a generative Naive Bayes classifier and a discriminative SVM classifier, to form a new algorithm (Co-NB-SVM). A similar approach was taken Chen et al. (2017), who combined 2 SVM classifiers into one.

3.1.2.2 Tri-training (Single algorithm, single view)



Tri-training is a variant of the co-training algorithm, which was introduced to fill in the gap where the *independent and redundant* assumption required for co-training cannot be fulfilled. In most scenarios, it is difficult to meet such a requirement. Nigam et al. (2000) has proven that there is a strong dependence on this assumption of independent and redundant feature split for co-training performance. In contrast, Tri-training can be easily applied to most data mining situations. (M. Li & Z.H. Zhou, 2005) It works by first bootstrap sampling the original labelled dataset into multiple bootstrap samples, and then training 3 initial classifiers with these bootstrap samples.

3.1.2.3 Democratic-Co (Multiple algorithms, single view)



✓
Can be applied more broadly than Co-Training
Few training data (labelled)
1. Trains classifiers
NB Classifier
KNN Classifier
C4.5 Classifier
2. Predict label
Many unlabelled data
Weighted voting, majority selected as predicted label
Many labelled data
3. Predicted output
4. Retrain classifiers with new labelled examples

Approaches such as Tri-training and Co-Forest that require bootstrapping, however, may only be successful if there is a significant number of labelled data. (K. Nigam & R. Ghani, 2000) Another approach to conducting co-training when the independent and redundant assumption cannot be fulfilled is to use an ensemble of different learning algorithms. As different learning algorithms have different inductive bias, co-training two classifiers with different algorithms with the same dataset can successfully improve the accuracy of the learned hypothesis. (Zhou & Goldman, 2000)

Statistical-Co was first proposed by Zhou & Goldman in 2000, where two different SL algorithms are first used to partition the input space into a set of equivalence classes, and the predictions from these algorithms are then combined through statistical methods to a produce a final decision. They then continued to build on their earlier work by introducing Democratic-Co (Zhou & Goldman, 2004), which combined outputs of classifiers using weighted voting instead of statistical methods. This improved algorithm was even more applicable to most situations, as it eliminated the need for statistical tests and partitioning into equivalence classes.

3.2 Support Vector Machine (SVM)

SVM-based methods makes use of the cluster assumption and infers that low-density areas are optimal areas for decision boundaries. One such algorithm is the Transductive Support Vector Machine (TSVM) that was proposed by Joachims et al. (1999). Instead of just using labelled data to train the SVM model, TSVM uses both unlabelled and labelled data by iteratively using the unlabelled data as the test dataset for the original SVM model. Since then, alternative SVM-based approaches such as semi-supervised SVM (S3VM) and Laplacian SVM (LapSVM). LapSVM makes use of the manifold structure information of the unlabelled data by adding a Laplacian regularisation term into the objective function.

3.3 Graph-based

Graph-based SSC involves creating a weighted graph representing the dataset, where labeled and unlabeled instances are viewed as nodes, and the similarity between these instances are then reflected by weighted edges between them.

Zhu et al. utilized Gaussian random fields and harmonic functions (GFHF) on a k nearest neighborhood (kNN) graph, and predicted the labels of unlabeled samples via label propagation on this graph.

Zhou et al. proposed a local and global consistent (LGC) method on a completely connected graph under the assumption of consistency. Consistency is important to semi-supervised learning; it means nearby samples are more likely to have similar labels (local consistency) and samples in the same structure are likely to share the same label (global consistency) (Zhou et al., 2003).

Belkin et al. took advantages of manifold learning and introduced a regularization framework—manifold regularization (MR). Both GFHF and LGC are transductive classifiers, which cannot predict unseen samples explicitly. On the contrary, MR is inductive and therefore can infer unseen samples directly

Wang et al. brought forward a semi-supervised classification method based on linear neighborhood propagation (LNP).

3.4 Neural Network Models (Deep SSL)

Much of the recent progress in machine learning is due to deep learning, as a result of better algorithms, better GPUs and availability of data. However, deep learning requires a lot of data does not work well for situations where we have a lot of data, but only a bit of them are labelled. Therefore, there has been increased interest in this area. A few techniques of note are Pseudo-labelling and Ladder Networks. However, most recently, Oliver et al. (2018) suggested that most deep SSL methods are based on benchmarks that are not able to address real world settings.

3.5 Safe Semi-Supervised Learning (S3L)

One new paradigm in the field of SSL is known as Safe Semi-Supervised Learning. The motivation behind the technique is due to the recognition that some unlabelled may be risky to the performance of the semi-supervised model. S3L improve traditional SSL by sieving out training instances that could degenerate the performance of the semi-supervised model.

3.6 Proposed Taxonomy

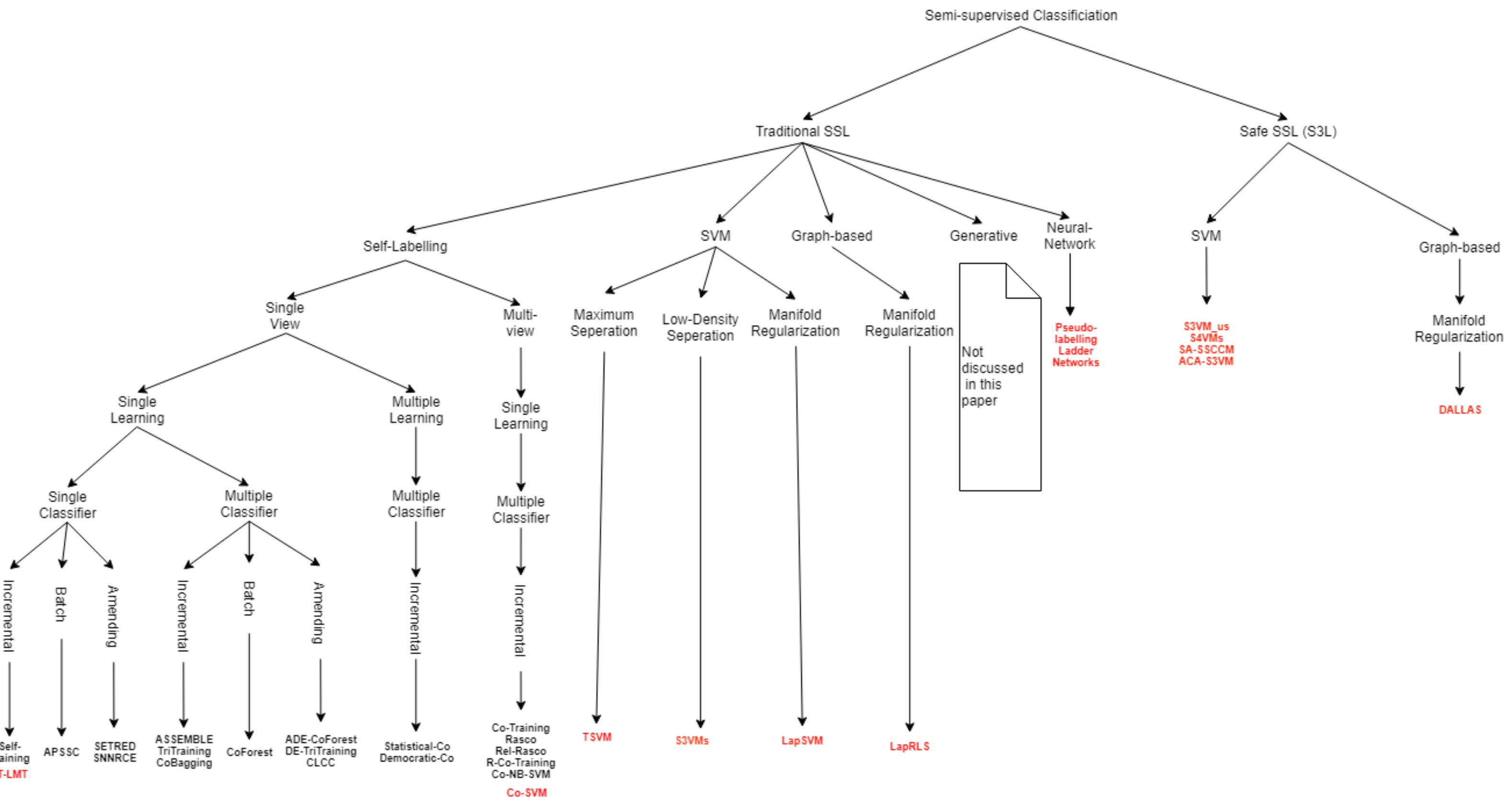

Semi-supervised Classificiation
Traditional SSL
Safe SSL (S3L)
Self-Labelling
SVM
Graph-based
Generative
Neural-Network
SVM
Graph-based
Single View
Multi-view
Maximum Seperation
Low-Density Seperation
Manifold Regularization
Manifold Regularization
Not discussed in this paper
Pseudo-labelling
Ladder Networks
S3VM_us
S4VMs
SA-SSCCM
ACA-S3VM
Manifold Regularization
DALLAS
Single Learning
Multiple Learning
Single Learning
Single Classifier
Multiple Classifier
Multiple Classifier
Multiple Classifier
Incremental
Batch
Amending
Incremental
Batch
Amending
Incremental
Incremental
Self-
aining
T-LMT
APSSC
SETRED
SNNRCE
ASSEMBLE
TriTraining
CoBagging
CoForest
ADE-CoForest
DE-TriTraining
CLCC
Statistical-Co
Democratic-Co
Co-Training
Rasco
Rel-Rasco
R-Co-Training
Co-NB-SVM
Co-SVM
TSVM
S3VMs
LapSVM
LapRLS

Figure 1. Proposed taxonomy

4. DUAL LEARNING-BASED SAFE SSL (DALLAS)

As introduced above, one state-of-the-art SSL method is the DuAL Learning-based sAfe Semi-supervised learning (DALLAS) which utilises dual learning to estimate the safety or risk of the unlabeled instances. In this section, we will study DALLAS more deeply and look at areas where its application can be extended.

4.1 DALLAS Motivation

As SSL involves the algorithm making use of unlabelled data as part of its training to make prediction., there are 3 parts to the SSL model which will affect its final prediction. As illustrated in Figure 1, they are namely

(1) the initial training labelled data that we choose to train the SSL model on
(2) the type of supervised classifier we use for prediction
(3) the unlabelled training data the SSL model is exposed to.

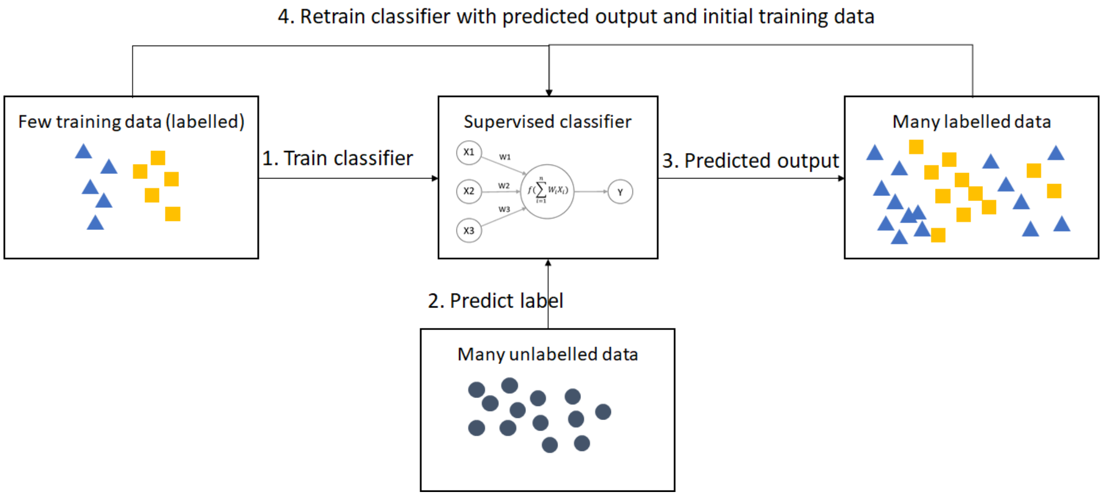


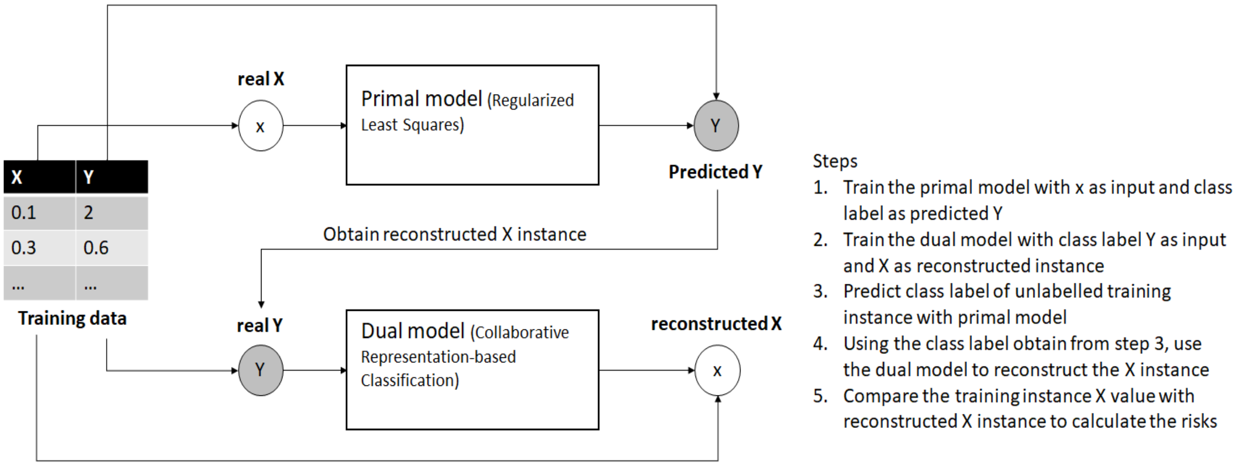
Figure 2: Traditional semi-supervised learning model

The initial labelled training data for which the SSL model is to be trained on has to be representative of the problem domain. This is important because unlabelled data can only be used for training purposes if they are labelled accurately. This further highlight the fundamental need to classify labelled data based on their risk level to optimise the performance of SSL. DALLAS proposed a method to safely exploit the use of unlabelled data for training using a dual learning-based approach to estimate risk of unlabelled training instance.

4.2 DALLAS Rationale

In DALLAS (Gan, H. et al, 2017), it was suggested that a primal model m1 that uses only labelled data is to be constructed to predict its class value. A dual model m2 that attempts to predicts the underlying features' value of the training instance through its class value is also required to be constructed. The output of the dual model will provide us with the reconstructed feature value for a typical label that we feed to the m2 model.

By comparing the L2 norm distance of the feature values of the unlabelled data instance with the predicted reconstructed instance for that same data instance, we will be able to calculate the risk associated with that data instance. In summary, the following diagram shows how a primal and dual classifier might calculate the SSL risks of each data instance.


real X
x
Primal model (Regularized Least Squares)
Y
Predicted Y
Obtain reconstructed X instance
real Y
Y
Dual model (Collaborative Representation-based Classification)
reconstructed X
x
Training data

X	Y
0.1	2
0.3	0.6
...	...

Steps

1. Train the primal model with x as input and class label as predicted Y
2. Train the dual model with class label Y as input and X as reconstructed instance
3. Predict class label of unlabelled training instance with primal model
4. Using the class label obtain from step 3, use the dual model to reconstruct the X instance
5. Compare the training instance X value with reconstructed X instance to calculate the risks

Figure 3: Summary of primal and dual model in calculating SSL risk of training data

As shown in the above diagram, the primal model to be used is Regularized Least Squares (RLS). The dual model used is Collaborative Representation-based Classification (CRC).

With the knowledge of which training instance are of high SSL risk, the following predictive model decisions can be made.

- Include only safe unlabelled instance (SSL < risk threshold) as training to the unsupervised portion of SSL model
- Spend effort to obtain labels of unsafe training instances (SSL > risk threshold) so that they could be use as supervised learning in SSL model
- Use the calculated risk as a penalty term to reduce overfitting for unsafe training instance

5. PROPOSED MODEL

In the DALLAS paper, Laplacian RLS was used as their SSL algorithm. The risk evaluation methodology proposed was only used to derive a risk-based regularisation term which is added into the objective function of Laplacian RLS. In this section, a novel SSL model is proposed, leveraging on risk-classification from DALLAS and traditional supervised algorithm. Given a mix of labeled and unlabeled data, the model aims to be able to make accurately prediction by following the following 3 key steps:

(1) Classify "risky" unlabeled data sets from "safe" unlabeled data set using a risk classification method implemented in DALLAS.
(2) Use Regularised Least Squares (RLS) model that was trained on the labelled data to predict the label of the "safe" unlabelled data.
(3) Run traditional SL algorithm on the initial labeled data and self-labelled output of the safe unlabelled data.

It is important to note that step 3 works under the assumption that the prediction of the RLS model in (2) on the "safe" unlabelled data is always true. The following section will explore the detailed implementation of our proposed method.

5.1 Risk-Based Classification of Unlabeled Data

Borrowing the concept from DALLAS, the risk-based classification of the unlabeled data sets is implemented in the following key steps:

Step (i)

Given a labelled training dataset, we need to first train a primal classifier using the regularised least square regression of the following equation.

$$\sum_{i=1}^{l} \quad (f(x_i) - y_i)^2 + \gamma || \, f \, ||_2^2 \qquad (1)$$

Where γ is a regularization parameter.

Step (ii)

Following that, we will need to train the dual model using Collaborative Representation-based Classification (CRC) which will reconstruct y from x. The equation is given as follows.

$$\widehat{a_k} \ = (X^T X + \beta I)^{-1} (X^T) \, y \qquad (2)$$

Where X is the matrix representation of the training labelled data set, and I is the identity matrix, β is the regularization parameter and y is the corresponding matrix representation class label of the X. Multiplying a $x_i$vector with $\widehat{a_k}$allow us to reconstruct its y value.

Step (iii)

For each instance of y, we will calculate the L2 norm distance of the standard deviation across the X feature space. This can be calculated as follows for each y instance.

$$\sigma_i \ = ||X_1^\sigma + X_2^\sigma + X_2^\sigma + \ldots + X_n^\sigma||_2^2 \quad (3)$$

Step (iv)

Using the model from equation 1, we predict $\widehat{y}_i$given $x_i = X$. After calculating $\widehat{y}_i$, we use the model from equation 2 to reconstruct $\widehat{y}_i$into $X_{\widehat{y}_i}$. We then use back equation 1 to predict $y_{X_{\widehat{y}_i}}$ given $X_{\widehat{y}_i}$.

Finally, we can calculate riskiness of $x_i = X$ using the following equation.

$$if\ \widehat{y}_i = y_{X_{\widehat{y}_i}}$$

$$r_i = exp\{-\frac{||X - X_{\widehat{y}_i}||_2^2}{2{\sigma_i}^2}\}$$

$$else \qquad (4)$$

$$r_i = exp\{\frac{||X - X_{\widehat{y}_i}||_2^2}{2{\sigma_i}^2}\}$$

Where r_i is the calculated risk and σ_i should be the σcalculated from equation 3 for the label $y = \widehat{y}_i$ e.g. $\sigma_{\widehat{y}_i}$.

Step (v)

Select the risk threshold t in which unlabelled data are considered safe when $r_i \le t$.

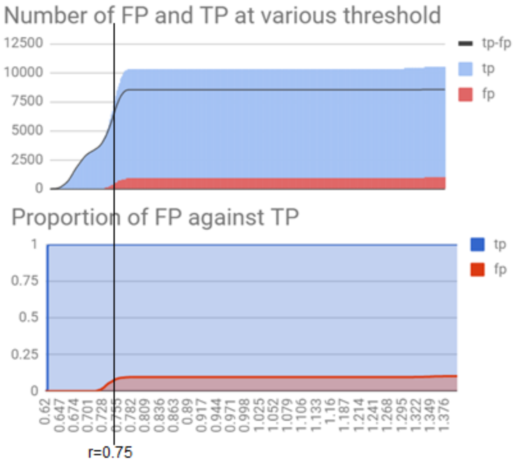


Figure 4: Risk Threshold Selection

The risk threshold $t$chosen was based on the following criterias.

(1) The growth of number of true positive in proportion to false positive. For example, we can see from the top graph in the above figure, when t reaches 0.75, the number of of false positive

would increase by 1.56x when t = 0.78 (580 to 909), while number of true positive would only increase by about 1.24x (7624 to 9428)

(2) Number of true positive must be > 7000

(3) Number of false positive must be < 600

5.2 Prediction of the Label of the "Safe" Unlabelled Data

To predict the label of the "safe" unlabelled data, we use back the Regularised Least Squares (RLS) model that was trained on the labelled data to predict the label of the "safe" unlabelled data as mentioned in section 5.1 step (iv). This allows us to reconstruct Y from X and predict the label of the "safe" unlabelled data.

$$if\ \hat{y_i} = y_{X_{\hat{y_i}}}$$

$$r_i = exp\{-\frac{||X - X_{\hat{y}_i}||_2^2}{2{\sigma_i}^2}\}$$

$else$

$$r_i = exp\{\frac{||X - X_{\hat{y}_i}||_2^2}{2{\sigma_i}^2}\}$$

5.3 Run Supervised Algorithm

At this point, the entire data set has 2 subsets:

(A) Initial labeled data set

(B) Self-labelled output for the "safe" unlabelled data set

As explained previously, data subset (B) is generated using the RLS model that was trained previously, as explained in section 5.1. The "risky" unlabeled data set has been removed.

With this data set, we run various traditional supervised algorithm, with the key assumption that the self-labelled output for the "safe" unlabelled data set is 100% true. This is done using an open-source SL module provided by the Knowledge Extraction based on Evolutionary Learning (KEEL) software tool, a research tool that contains a wide range of machine learning algorithms[1].

In the next section, the performance of the overall SSL model is then evaluated, using raw dataset of 10% labeled data and 90% unlabeled data.

[1] Alcalá-Fdez J, Sánchez L, García S, del Jesus MJ, Ventura S, Garrell JM, Otero J, Romero C, Bacardit J, Rivas VM, Fernández JC, Herrera F (2009) KEEL: a software tool to assess evolutionary algorithms for data mining problems. Soft Comput 13(3):307–318

6. EXPERIMENTAL ANALYSIS

6.1 Dataset

The benchmark dataset used to train and measure the performance of our SSL model comes from the following link http://sci2s.ugr.es/keel/semisupervised.php. The name of the dataset zip file is called nursery-ssl10 which is already partitioned by means of 10-folds cross validation procedure. However, only the first fold of the partitioned dataset are used for measuring the performance of our model i.e. we are using the csv dataset file containing the name nursery-ssl10-10-1 as the raw data for our analysis.

(1) nursery-ssl10-10-1-tra.csv
This csv dataset file contains 10% labelled training data and 90% unlabelled training data. This dataset is used to train the initial SSL model and allow it to perform labelling for the 90% of unlabelled data. The SSL model will then be retrained on the dataset which it has just labelled.

(2) nursery-ssl10-10-1-trs.csv
This csv dataset contains all the labelled class information which is not available in the nursery-ssl10-10-1-tra.csv dataset. In other words, it is the answers key for the unlabelled data instances in training dataset. We use this to confirm the quality of our SSL model in labelling the unlabelled training data.

(3) nursery-ssl10-10-1-tst.csv
This csv dataset is the test dataset used to measure the accuracy of the SSL model implemented.

Appendix 1 describes the data set names for different purposes in the experiments that we will be conducting in the next few sections in details.

6.2 Algorithms Used

In the comparison analysis in the next few sections, various SL and SSL algorithm are utilised by means of the KEEL software tool. *Appendix 2* summarises the various algorithms used and their sources.

6.3 Experiment 1: Proposed Model vs Traditional SSL Method

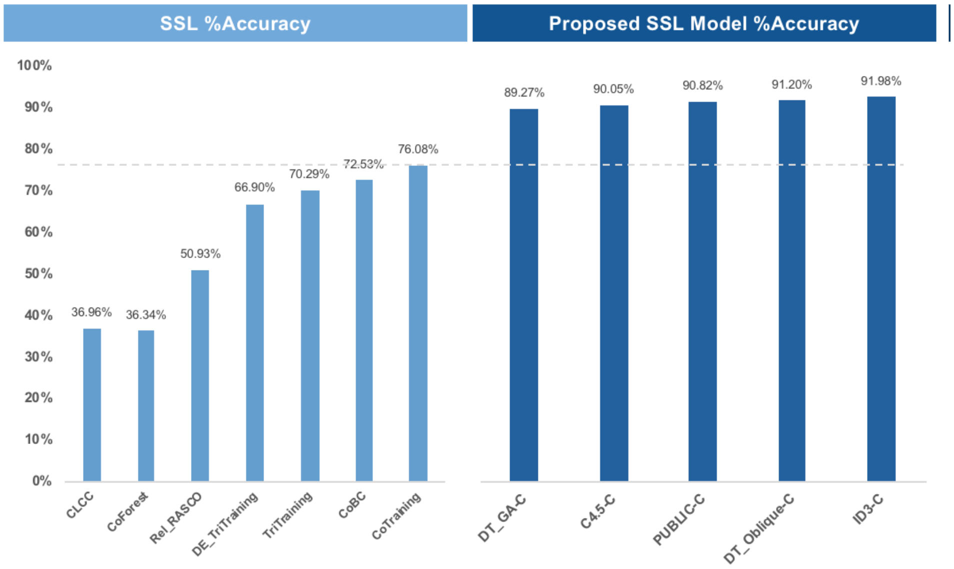


Figure 5: %Accuracy of Traditional SSL and Proposed SSL Model

Method	Traditional SSL	Proposed SSL Model
Algorithm	CLCC, CoForest, Rel_RASCO, DE_TriTraining, TriTraining, CoBC, CoTraining	DT_GA-C, C4.5-C, PUBLIC-C, DT_Oblique-C, ID3-C
Training Data Set	nursery-ssl10-10-1tra.dat (11664 data) nursery-ssl10-10-1trs.dat	nursery-safe-label-tra.dat (9371 data)
Test Data Set	nursery-ssl10-10-1tst.dat (identical)	nursery-sl-tst.dat (identical)

Table 1: Experiment 1 Details

When comparing the test results of our proposed model to the traditional SSL algorithm which does not include classification of the “risky” unlabeled data from the “safe” ones, the our proposed SSL model outperforms all traditional SSL algorithms. The lowest accuracy of the proposed model is 89.27% (DT_GA-C), which is higher than the top-performing Co-Training algorithm at 76.08%.

This experiment shows that our proposed SSL model shows promising results as its accuracy is significantly higher than the traditional SSL algorithm.

6.4 Experiment 2: Proposed Model vs SL (Complete)

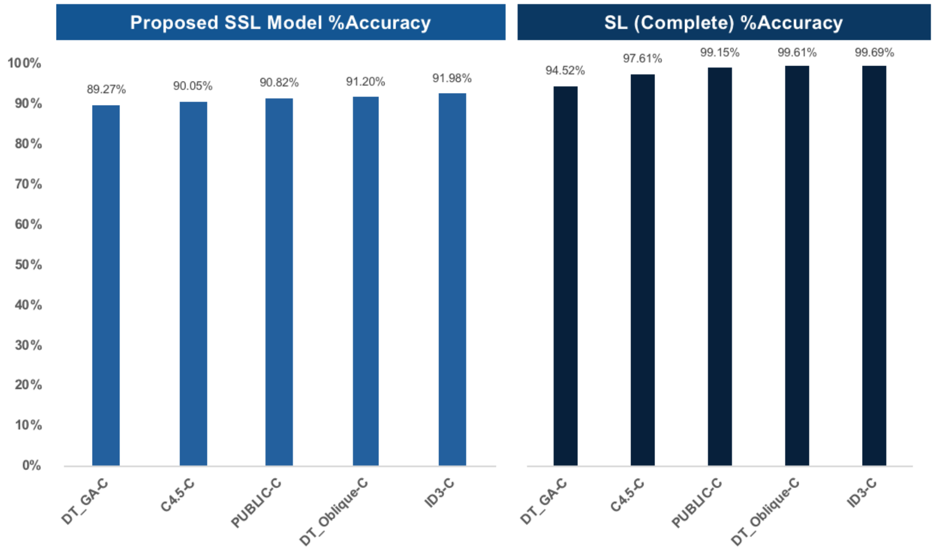


Figure 6: %Accuracy of Proposed SSL Model and SL (Complete)

Method	Proposed SSL Model	SL (Complete)
Algorithm	DT_GA-C, C4.5-C, PUBLIC-C, DT_Oblique-C, ID3-C	DT_GA-C, C4.5-C, PUBLIC-C, DT_Oblique-C, ID3-C
Training Data Set	nursery-safe-label-tra.dat (9371 data)	nursery-sl-tra.dat (11664 data)
Test Data Set	nursery-sl-tst.dat (identical)	nursery-sl-tst.dat (identical)

Table 2: Experiment 2 Details

SL (Complete) uses the complete 100% labeled data set from the raw file *nursery-ssl10-10-1-tra.csv*. As such, the data is 100% true as it is, as compared to our proposed model that has predicted labels on the safe unlabeled subset. Despite this, our model's performance is very close to that of the fully supervised model using the complete data sets of 11,664 data.

The lower performance of our proposed SSL model may be attributed to 2 things. Firstly, the wrong prediction output of the safe unlabeled data as the assumption that all predictions are 100% correct may not hold true. Secondly, it may also be attributed to the smaller training data set of 9371, which is around 19.7% smaller than that of the raw data set used in SL (Complete).

6.5 Experiment 3: Proposed Model vs SL (Safe Answer)

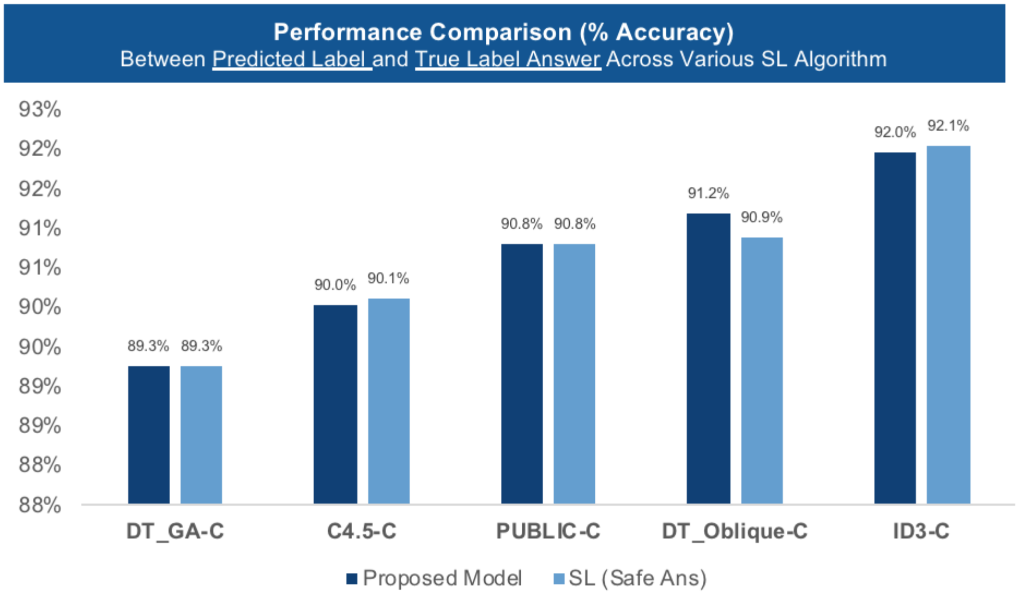


Figure 7: %Accuracy of Proposed SSL Model and SL (Safe Ans)

Method	Proposed SSL Model	SL (Safe Ans)
Algorithm	DT_GA-C, C4.5-C, PUBLIC-C, DT_Oblique-C, ID3-C	DT_GA-C, C4.5-C, PUBLIC-C, DT_Oblique-C, ID3-C
Training Data Set	nursery-safe-label-tra.dat (9371 data)	nursery-safe-ans-tra.dat (9371 data)
Test Data Set	nursery-sl-tst.dat (identical)	nursery-sl-tst.dat (identical)

Table 3: Experiment 3 Details

The main difference between the two data sets is in the training data set used. SL (Safe Ans) uses the data set that contains the true answer of the safe unlabelled data, as compared to a prediction in our proposed SSL model.

This experiment shows that overall, the assumption that 100% of the output prediction of the safe unlabelled data is correct is justified as seen in the tiny performance difference shown in this experiment. This highlights that our method of prediction using RLS is highly accurate as the performance is on par to that of SL (Safe Ans), even outperforming in the case of DT_Oblique-C algorithm.

6.6 Overall Result

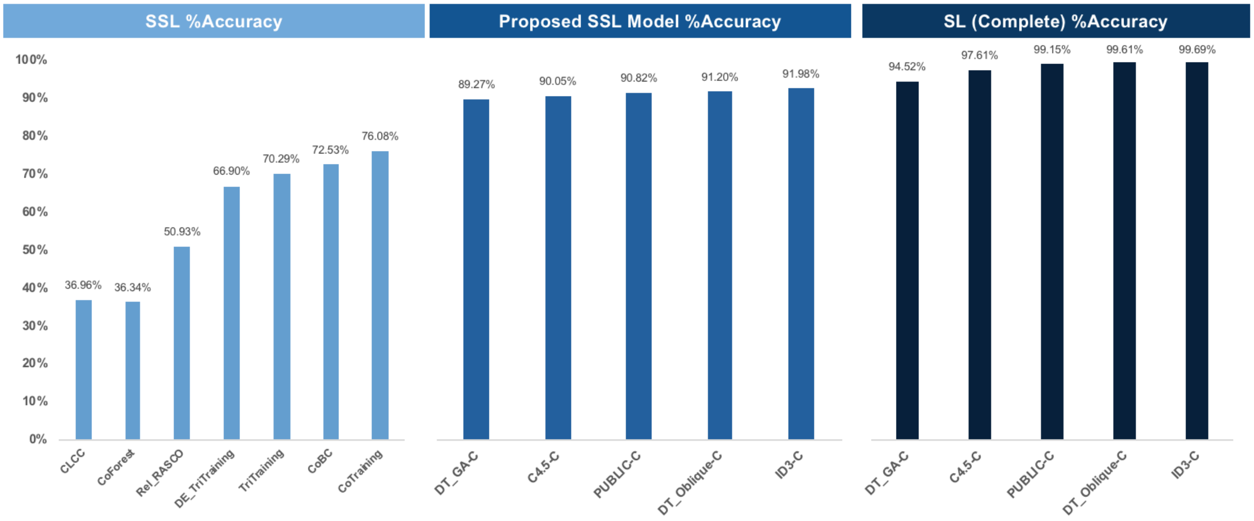


Figure 8: %Accuracy of Traditional SSL, Proposed SSL Model and SL (Safe Ans)

As seen in the overall result in Figure 7, our model's performance sits in between that of the traditional SSL method and the fully supervised algorithm. Our SSL model outperforms all the traditional SSL algorithms and has the lowest performance of 89.27% accuracy. While the proposed SSL model's performance is lower than that of SL (Complete), the performance difference is minimal, with an average difference of -7.45%.

6.7 Limitations and Mitigation

There may be some limitations involved in the use of the proposed SSL method in this paper.

The use of dual learning in assessing riskiness of unlabelled dataset is affected by the curse of dimensionality for both the primal model and the dual model. When number of feature space in X is much more than Y, then reconstruction from Y to X will not be accurate. As such, to reduce the dimensionality of data, we can consider only the important features by using techniques such as principal component analysis. This helps minimise the limitation identified earlier and as shown in the result of Experiment 3 in section 6.5, the reconstruction from Y to X is highly accurate.

Moreover, other limitations include the risk of inefficiency of dual learning if data used are of high dimension. In other words, when the data have more features than classes, reconstruction of Y to X may not be accurate as well. This can be overcome by reducing dimensionality of data by considering only the important features. Alternatively, one can also mitigate this limitation by moving some of the features as outcome.

7. Conclusion

In this paper, we have two key objectives. Firstly, to explore the existing and state-of-the-art SSL methods, we conduct in-depth literature review of various papers and and classify them into a SSL taxonomy that we propose. Secondly, to propose an extension of the application of risk-based classification in DALLAS, we propose a novel SSL model that utilises risk-classification of unlabeled data, RLS prediction and traditional supervised algorithm. To verify the accuracy of the proposed model, several rounds of experiments are conducted. Upon rigorous evaluation of the proposed SSL model, it is revealed that the model is able to outperform the traditional SSL algorithm and is able to perform slightly below par to fully supervised method using 100% labeled data.

Certainly, there are some research directions that can be further explored. For instance, the application of risk-based classification and RLS prediction on unlabelled data may be extended to other algorithms not explored in this paper.

References

A. Blum, T. Mitchell. (1998). Combining labeled and unlabeled data with co-training. Proceedings of the annual ACM conference on computational learning theory, 92-100. Retrieved from: https://www.cs.cmu.edu/~avrim/Papers/cotrain.pdf

A. Halder, S. Ghosh, A. Ghosh. (2013). Aggregation pheromone metaphor for semi-supervised classification. Pattern Recognition, 46. 2239-2248.

D.Y. Zhou, O. Bousquet, T.N. Lal, J. Weston, B. Scholkopf, Learning with local ¨ and global consistency, in: Proceedings of the Advances in Neural Information Processing Systems (NIPS), 2003.

D.Y. Zhou, O. Bousquet, T.N. Lal, J. Weston, B. Scholkopf, Learning with local and global consistency, in: Proceedings of the Advances in Neural Information Processing Systems (NIPS), 2003.

D. Yarowsky. (1995). Unsupervised word sense disambiguation rivaling supervised methods. 33rd Annual Meeting of the Association for Computational Linguistics, 189-196. Retrieved from: http://www.aclweb.org/anthology/P95-1026

F.G. Cozman, I. Cohen, M.C. Cirelo, (2003). Semi-supervised learning of mixture models, in: Proceedings of International Conference on Machine Learning,Washington, DC, pp. 41–65.

Gan, H., Li, Z., Fan, Y., & Luo, Z. (2017). Dual Learning-Based Safe Semi-Supervised Learning. IEEE Access, 6, 2615-2621. doi:10.1109/access.2017.2784406

I. Triguero, S. García, & F.Herrera (2015). Self-Labeled Techniques for Semi-Supervised Learning: Taxonomy, Software and Empirical Study. Knowledge and Information Systems, 42, 245-284. Retrieved from: http://sci2s.ugr.es/keel/papers/semi-supervised/2013-KAIS-Triguero.pdf

J.D. Wang, F. Wang, C.S. Zhang, H.C. Shen, L. Quan, Linear neighborhood propagation and its applications, IEEE Transactions on Pattern Analysis and Machine Intelligent 31 (9) (2009) 1600–1615.

J. Wang, S. Luo, X. Zeng. (2008). A random subspace method for co-training. IEEE international joint conference on computational intelligence, 195-200. doi:10.1109/IJCNN.2008.4633789

K. Nigam, A.K. McCallum, S. Thrun, T. Mitchell, (2000). Text classification from labeled and unlabeled
documents using EM, Mach. Learn. 39 (2–3), 103–134.

K. Nigam, R. Ghani. (2000) Analyzing the effectiveness and applicability of co-training. Proceedings of the 9th International Conference on Information and Knowledge Management. 86–93. Retrieved: http://www.kamalnigam.com/papers/cotrain-CIKM00.pdf

M. Li, Z.H. Zhou. (2005). SETRED: self-training with editing. Advances in Knowledge Discovery and Data Mining. LNCS 3518, Springer 2005611-621. Retrieved from: https://cs.nju.edu.cn/zhouzh/zhouzh.files/publication/pakdd05a.pdf

M. Li, Z.H. Zhou. (2005). Tri-training: exploiting unlabeled data using three classifiers. IEEE Transactions on Knowledge and Data Engineering, 17, 1529-1541. doi:10.1109/TKDE.2005.186

M. Belkin, P. Niyogi, V. Sindhwani. (2006). Manifold regularization: a geometric framework for learning from labeled and unlabeled examples, Journal of Machine Learning Research 7, 2399–2434.

Nigam, K., McCallum, A.K., Thrun, S., Mitchell, T. (2000). Text classification from labeled and unlabeled documents using EM. Machine Learning, 39(2), 103–134. doi:10.1023/A:1007692713085

O. Chapelle, A. Zien. (2005). Semi-supervised classification by low density separation. AISTATS, 57–64.

S. Dasgupta, M. Littman, D. McAllester. (2002). PAC Generalization Bounds for Co-Training. Advances in Neural Information Processing Systems, 14. 375-382. Retrieved from: http://cseweb.ucsd.edu/~dasgupta/papers/cotrain.pdf

S. Goldman, Y. Zhou. (2000). Enhancing supervised learning with unlabeled data. Proc. of the 17th Int. Conf. on Machine Learning, 327-334. Retrieved from:http://www.cs.columbia.edu/~dplewis/candidacy/goldman00enhancing.pdf

S. Karlos, A. Panagopoulou, N. Fazakis, N., Sgarbas, K.)., & Kotsiantis, S. (2017). Locally application of naive Bayes for self-training. Evolving Systems, 8(1), 3-18. doi:10.1007/s12530-016-9159-3

T. Huang, Y. Yu, G. Guo, K. Li. (2013). A classification algorithm based on local cluster centers with a few labeled training examples. Knowledge Based Systems 26:6, 563-571.

T. Joachims. (1999). Transductive inference for text classification using support vector machines. Proceedings of International Conference on Machine Learning,Slovenia, pp. 200–209.

X.J. Zhu, Z. Ghahramani, J. Lafferty.(2003). Semi-supervised learning using Gaussian fields and harmonic functions, in: Proceedings of the 20th International Conference on Machine Learning (ICML), 2003.

Y. Chen, T. Pan and S. Chen. (2017). Development of co-training support vector machine model for semi-supervised classification. 2017 36th Chinese Control Conference (CCC), Control Conference (CCC), 2017 36th Chinese, 11077. doi:10.23919/ChiCC.2017.8029125

Y. Grandvalet, Y. Bengio. (2004). Semi-supervised learning by entropy minimization. Proceedings of Advances in Neural Information Processing Systems, 529–536.

Y. Wang, X. Xu, H. Zhao, Z. Hua. (2010). Semi-supervised learning based on nearest neighbor rule and cut edges. Knowledge-Based Systems, 23:6, 547-554. doi:10.1016/j.knosys.2010.03.012

Y. Yaslan, Z. Cataltepe. (2010). Co-training with relevant random subspaces. Neurocomputing, 73,1652-1661. doi:10.1016/j.neucom.2010.01.018

Y. Zhou, S. Goldman. (2004). Democratic co-learning. IEEE international conference on tools with artificial intelligence, 594-602. doi:10.1109/ICTAI.2004.48

Appendix 1: Datasets Specifications

Proposed Model		
Name	Data	Number of Data
nursery-safe-label-tra.dat	Training data set containing initial labeled data, output of the prediction of safe unlabeled data.	9371
nursery-sl-tst.dat	Test data set	1296
SL Complete		
nursery-sl-tra.dat	Training data set containing the answer of the original training data set. Identical to nursery-ssl10-10-1-trs.dat	11664
nursery-sl-tst.dat	Test data set	1296
SL Safe Answer		
nursery-safe-ans-tra.dat	Training data set containing the answer of the safe data set only. It contains the true answer of the safe unlabelled data sets.	9371
nursery-sl-tst.dat	Test data set	1296
SSL (Traditional)		
nursery-ssl10-10-1tra.dat	Raw training data set for SSL. Contains 10% labeled and 90% unlabeled data set.	11664
nursery-ssl10-10-1trs.dat	Answer data set that contains all the labelled class information that are not available in nursery-ssl10-10-1tra.dat.	11664
nursery-ssl10-10-1tst.dat	Test data set, identical to nursery-sl-tst.dat	1296

Table 3: Datasets Specifications

Appendix 2: Algorithms

SSL ALGORITHM		
Abbreviation	**Full Name**	**Reference**
CLCC	CLCC	T. Huang, Y. Yu, G. Guo, K. Li. A classification algorithm based on local cluster centers with a few labeled training examples. Knowledge Based Systems 26:6 (2010) 563-571.
CoBC	CoBC	M. Hady, F. Schwenker, G. Palm. Semi-supervised learning for tree-structured ensembles of RBF networks with Co-Training. Neural Networks 23 (2010) 497-509.
CoForest	CoForest	M. Li, Z.H. Zhou. Improve Computer-Aided Diagnosis With Machine Learning Techniques Using Undiagnosed Samples. Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on 37:6 (2007) 1088-1098.
CoTraining	CoTraining	A. Blum, T. Mitchell. Combining labeled and unlabeled data with co-training. Proceedings of the annual ACM conference on computational learning theory. (1998) 92-100.
DE_TriTraining	Differential Evolution TriTraining	C. Deng, M. Guo. Tri-training and data editing based semi-supervised clustering algorithm. MICAI 2006: Advances in Artificial Intelligence (MICAI 2006). (2006) 641-651.
Rel_RASCO	Co-training with relevant random subspaces	Y. Yaslan, Z. Cataltepe. Co-training with relevant random subspaces. Neurocomputing 73 (2010) 1652-1661.
TriTraining	TriTraining	Z.H. Zhou, M. Li. Tri-training: exploiting unlabeled data using three classifiers. IEEE Transactions on Knowledge and Data Engineering 17 (2005) 1529-1541.

Table 4: SSL Algorithms

SL ALGORITHM (DECISION TREES)		
Abbreviation	**Full Name**	**Reference**
C4.5-C	C4.5	J.R. Quinlan. C4.5: Programs for Machine Learning. Morgan Kauffman, 1993.
ID3-C	Iterative Dicotomizer 3	J.R. Quinlan. Induction of Decision Trees. Machine Learning 1 (1986) 81-106.
DT_GA-C	Hybrid Decision Tree - Genetic Algorithm	D.R. Carvalho, A.A. Freitas. A hybrid decision tree/genetic algorithm method for data mining. Information Sciences 163:1 (2004) 13-35.
DT_Oblique-C	Oblique Decision Tree with Evolutionary Learning	E. Cantú-Paz, C. Kamath. Inducing oblique decision trees with evolutionary algorithms. IEEE Transactions on Evolutionary Computation 7:1 (2003) 54-68.
PUBLIC-C	PrUning and BuiLding Integrated in Classification	R. Rastogi, K. Shim. PUBLIC: A Decision Tree Classifier that Integrates Building and Pruning. Data Mining and Knowledge Discovery 4:4 (2000) 315-344.

Table 5: SL Algorithm (Decision Trees)