Guided Self-Training based Semi-Supervised Learning for Fraud Detection

Awanish Kumar
Soumyadeep Ghosh
Janu Verma
AI Garage, Mastercard
India

ABSTRACT

Semi supervised learning has attracted attention of AI researchers in the recent past, especially after the advent of deep learning methods and their success in several real world applications. Most deep learning models require large amounts of labelled data, which is expensive to obtain. Fraud detection is a very important problem for several industries and large amount of data is often available. However, obtaining labelled data is cumbersome and hence semi-supervised learning is perfectly positioned to aid us in building robust and accurate supervised models. In this work, we consider different kinds of fraud detection paradigms and show that a self-training based semi-supervised learning approach can produce significant improvements over a model that has been training on a limited set of labelled data. We propose a novel self-training approach by using a guided sharpening technique using a pair of autoencoders which provide useful cues for incorporating unlabelled data in the training process. We conduct thorough experiments on three different real world databases and analysis to showcase the effectiveness of the approach. On the elliptic bitcoin fraud dataset, we show that utilizing unlabelled data improves the F_1 score of the model trained on limited labelled data by around 10%.

CCS CONCEPTS

• **Security and privacy → Authentication**.

KEYWORDS

adversarial attack, vulnerability detection, vulnerability mitigation, transaction level vulnerability, black box vulnerability detection

ACM Reference Format:
Awanish Kumar, Soumyadeep Ghosh, and Janu Verma. 2022. Guided Self-Training based Semi-Supervised Learning for Fraud Detection. In *3rd ACM International Conference on AI in Finance (ICAIF '22),November 2–4, 2022, New York, NY, USA.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3533271.3561783

ICAIF '22, November 2–4, 2022, New York, NY, USA

ACM ISBN 978-1-4503-9376-8/22/11…$15.00
https://doi.org/10.1145/3533271.3561783

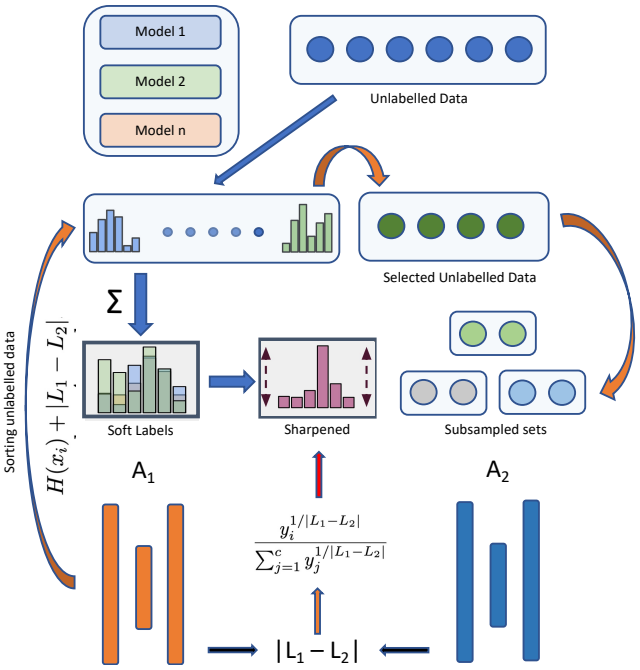

Figure 1: Illustrating the concept of the proposed approach.

1 INTRODUCTION

Real world problems such as fraud detection are both contemporary and pertinent to the financial, e-commerce, social media, and other industries. These platforms have in their hands an enormous amount of data, however labelling the data is extremely expensive and time-consuming. Modern machine learning algorithms although yield high accuracies, require large amounts of labelled data to provide a consistent level of performance. In some cases, even if we have large amounts of labelled data, it has been shown that incorporating additional data, either labelled or unlabeled, into the model training process, may result in a better model performance. Thus, utilizing the recent progress of semi-supervised learning in fraud detection problems should help us in building better and more reliable models.

With the advent of a variety of online applications such as payment methods, hotel booking, e-commerce websites, food delivery services and so on, fraudulent activities on such utilities have also been on the rise. As an example, an impostor on a payment network, a fake review on a hotel booking/rating website or food delivery/rating app may be encountered on a daily basis. These illegal or unethical activities needs to be detected and addressed since they result in the loss of a huge amount of monetary value

or goodwill for various companies and businesses. Such events in the technical term may be referred to as a fraud, which might have certain unique characteristics or attributes of its own. Machine learning methods are sought after to address this problem of modelling fraudulent activity by training models utilizing large quantities of data owned by these corporations who are at the receiving end of such fraud. However, most of the data is unlabelled and detected frauds are a very tiny fraction of such data repositories. Thus, this problem is challenging and most fraud detection methods has its own limitations, inadequacy of labelled data is at the heart of it.

Fraud detection needs to be addressed for several applications such as financial or credit card fraud (identity theft and merchant fraud), money laundering, loan fraud, insurance claim fraud, health care fraud (fake/overcharged bills), online reviewing fraud and so on. Addressing online review fraud is challenging due to the camouflage behaviour of fraudsters. In this work we focus on developing a general fraud detection method and exhibit its effectiveness on financial and online review frauds. Several methods [2–4, 11] have been proposed to detect fraud in online camouflaged fraudulent reviews. Most of these methods have harnessed the relationship of the reviews and the users, which can be modelled as graphs. The motivation for choosing such methods for fraud detection are (1) They allow unlabelled data to be utilized in training the models, which is abundantly available in such problems and, (2) The scope of utilizing multi-view data like social relations and general user/data attributes. The proposed semi-supervised method also takes into account these two aspects, as they are a generally fulfilled by most recent state-of-the-art methods for fraud detection.

1.1 Research Contributions

Fraud detection and analysis has been solved using both unsupervised and supervised machine learning models. Since fraud can also be considered as an anomaly or an outlier, such methods have also been utilized for the same. There are strong arguments for using both supervised and unsupervised paradigms of machine learning, however in this work we propose a supervised algorithm for it. We use an unsupervised model to guide the pseudolabelling process making this a hybrid solution for fraud detection. The reasons are twofold. Firstly a supervised approach allows us to utilize all the recent progress that has been made on semi-supervised learning from other domains such as computer vision and natural language processing. Secondly, all frauds are not outliers. Frauds are mostly novel data samples that have something peculiar in them and unsupervised learning algorithms may not allow us to exploit and learn such peculiarities, since such algorithms mostly create an understanding of the non-fraud data and then attempts to classify others as fraud. Thus, we build our proposed semi-supervised formulation based on supervised learning, along with an auxiliary unsupervised model guiding the semi-supervised learning process. The primary challenge in such a formulation is to utilize unlabelled data, which we perform using the classical process of pseudolabelling, where we assign labels to the unlabelled data samples using a model trained on limited training data. Thereafter, we select a set of unlabelled data points based on the confidence of the classifier pseudolabel for those samples. We utilize these samples along with the limited labelled data to train another model, which is expected to be am improved version of the previous model. This process is repeated until adding more unlabelled data yields no more benefit in terms of the model's performance on a fixed test data set. The research contributions of the work are as follows:

(1) We propose a guided self-training based semi-supervised learning method for fraud detection. The method can be used with a classification model, which utilizes unlabelled data iteratively using a student-teacher based pseudolabelling mechanism. We use an auxiliary autoencoder which gives us useful cues for improving the pseudolabelling process and prevent label noise. We also show that the proposed method can be used with any supervised learning model, not necessarily only with neural models.
(2) We exhibit the efficacy of the proposed method on two real-world fraud problems, namely online user reviews and financial fraud on bitcoin transactions. We show results on three databases namely, Amazon and Yelp reviews, and Eliptic Bitcoin dataset. The proposed method outperforms state-of-the-art recent fraud detection methods.

2 BACKGROUND

We consider a collection of n examples $X := \{x_1, x_2, \ldots, x_l, x_{l+1}, \ldots, x_n\}$ with $x_i \in \chi$. The first l examples $X_L := \{x_1, x_2, \ldots, x_l\}$ are labeled as $Y_L := \{y_1, y_2, \ldots, y_l\}$ where $y_i \in C$ a discrete set over c classes i.e. $C := \{1, 2, \ldots, c\}$. The remaining examples x_i for $i \in U := \{l+1, l+2, \ldots, n\}$ are unlabeled. If the unlabeled set is denoted as X_U , then X is the disjoint union of the two sets $X = X_L \cup X_U$. In supervised learning, we use the labeled examples with their corresponding labels (X_L, Y_L) to train a classifier that learns to predict class-labels for previously unseen examples. The guiding principle of semi-supervised learning is to leverage the unlabeled examples as well to train the classifier.

Supervised Learning: We assume a deep convolutional neural network (DCNN) based classifier trained on the labeled set of examples (X_L, Y_L) which takes an example $x_i \in \chi$ and outputs a vector of class-label probabilities i.e. $f_\theta : \chi \to \mathbb{R}^c$ where θ are the parameters of the model. The model is trained by minimizing the *supervised loss* -

$$L_s(X_L, Y_L, \theta) = \sum_{i=1}^{l} l_s(f_\theta(x_i), y_i) \tag{1}$$

A typical choice for the loss function l_s in classification is the *cross-entropy* $l_s(\hat{y}, y) = -y\log(\hat{y})$

A deep neural network can be thought of as the composition of two networks - *feature extraction network* which transforms an input example to a vector of features $\phi_\theta : \chi \to \mathbb{R}^d$ and *classification network* which maps the feature vector to the class vector. Let $v_i := \phi_\theta(x_i)$ be the feature vector of x_i. The classification network is usually a *fully-connected layer* on top of ϕ_θ. The output of the network for x_i is $f_\theta(x_i)$ and the final prediction is the class with highest probability score i.e

$$\hat{y}_i := \arg\max_j (f_\theta(x_i))_j \tag{2}$$

A trained classifier (at least the feature generator network) is the starting point of the most of the semi-supervised learning techniques, including the studies performed in this work.

Semi-supervised Learning (SSL): We use the following principle for introducing unlabelled data into the supervised training process.

- *Pseudo-labeling:* The unlabeled examples are assigned pseudo-labels thereby expanding the label set to all of X. A model is then trained on this labeled set $(X_L \cup X_U), (Y_L \cup \hat{Y}_U)$ using the supervised loss for the true-labeled examples plus a similar loss for the pseudo-labeled examples.

$$L_p(X_U, \hat{Y}_U, \theta) = \sum_{i=l+1}^{n} l_s(f_\theta(x_i), \hat{y}_i) \tag{3}$$

There are other methods for semi-supervised learning, the current work fits in the realm of the later school where we study the effect of iteratively adding pseudo-labeled examples for self-training.

Self Training using Student-Teacher Models for Semi-supervised Learning: This class of methods [10] for SSL iteratively use a trained (teacher) model to pseudo-label a set of unlabeled examples, and then re-train the model (now student) on the labelled plus the pseudo-labelled examples. Usually the same model assumes the dual role of the the student (as the learner) and the teacher (it generates labels, which are then used by itself as a student for learning). A model f_θ is trained on the labelled data X_L (using supervised loss equation 1), and is then employed for inference on the unlabeled set X_U. The prediction vectors $f_\theta(x_i) \forall x_i \in X'_U$ are converted to one-hot-vectors, where $X'_U \subset X_U$. These examples X'_U along with their corresponding (pseudo-)labels $\hat{Y}'_U$ are added to the original labelled set. This extended labelled set $X_L \cup X'_U$ is used to train another (student) model f'_θ. This procedure is repeated and the current student model is used as a teacher in the next phase to get pseudo-labels $\cup X''_U$ for training another (student) model f''_θ on the set $X_L \cup X'_U \cup X''_U$. Now, for conventional self-training methods we use the entire unlabeled set X_U in every iteration. However, as mentioned above, the most general form of self training can have different sets of unlabeled data (X'_U, X''_U and so on) in every iteration. The method of selection of X'_U from X_U can come from any utility function, the objective of which would be to use the most appropriate unlabeled data samples in each iteration. Some methods even use weights for each (labelled/unlabeled) data sample, which are updated in every iteration, similar to a process followed in Transductive Semi-Supervised Learning [7] methods, which is borrowed from the traditional concept of boosting used in statistics.

3 PROPOSED METHOD

In the following, we describe our recipe for self-training based semi-supervised learning. We assume that we have limited labelled data available, using which we train two autoencoders, namely A_1 and A_2 one for each class. We also train a supervised classification model M using the same labelled data as explained in the previous section. Then, we iteratively assign pseudo-labels to the unlabeled examples by using M and the autoencoders. The traditional self training method uses M to assign pseudolabels. The proposed approach also assigns pseudolabels by using the prediction vectors from M however, it selects a subset of unlabelled samples by using the reconstruction losses from A_1 and A_2. The main ingredients of the proposed approach are - autoencoders, sub-sampling, training, sample selection, pseudo-labeling, sharpening, and re-training. These are discussed below.

Training Autoencoders: As illustrated in Section 2, we consider a collection of n examples $X := \{x_1, x_2, \ldots, x_l, x_{l+1}, \ldots, x_n\}$ with $x_i \in \chi$. The first l examples $X_L := \{x_1, x_2, \ldots, x_l\}$ are labeled as $Y_L := \{y_1, y_2, \ldots, y_l\}$ where $y_i \in C$ a discrete set over c classes i.e. $C := \{1, 2, \ldots, c\}$. Since we are formulating the solution for a 2-class problem (fraud and non-fraud), (c=2 in this case) we train 2 autoencoders using the labelled data for each class.

Let X_F and X_{nF} be the unlabeled training data from two different (but registered) classes, namely fraud and non-fraud, respectively. Let $X_F = \left\{x_F^{(1)}, x_F^{(2)}, x_F^{(3)}, \ldots x_F^{(p)}\right\}$ be the training set comprising of fraud samples and $X_{nF} = \left\{x_{nF}^{(1)}, x_{nF}^{(2)}, x_{nF}^{(3)}, \ldots x_{nF}^{(q)}\right\}$ be the non-fraud samples, where $p + q = l$. Using these samples, we train A_1 and A_2 to learn the following mappings as is the usual norm with autoencoders, namely $\mathcal{A}_\infty : X_F \longrightarrow X_F$ and $\mathcal{A}_\in : X_{nF} \longrightarrow X_{nF}$.

Let the reconstruction loss for A_1 be L_1 and that of A_2 be L_2. Since A_1 is trained on fraud samples, it is likely to return a lower value of the reconstruction loss on fraud samples than non-fraud samples. Thus for an unlabelled sample x_i we can calculate both L_1 and L_2. If x_i is a fraud sample then it is likely that $L_1 < L_2$, else $L_1 > L_2$, in which case it is more likely to be a non-fraud sample. This is known as the *classification hint*. In addition to this, we may also calculate $|L_1 - L_2|$, the magnitude of which is directly proportional to the confidence of the system of autoencoders on the class of the sample. We would be using these cues during the pseudolabelling process for unlabelled data.

Sub-sampling: Let $T^{(i)} = (X^{(i)}, Y^{(i)})$ be a set of training examples where each $x_i \in X^{(i)}$ has a corresponding label $y_i \in Y^{(i)}$. We generate k random samples of the training set $T^{(i)}$ each of size m, $m < |T^{(i)}|$. An example can be present in more than one of the k samples i.e. we do not require $m * k$ to be equal to $|T^{(i)}|$.

Model Training: We train k separate models on each of the k samples of the training data. The models are also chosen to be different architectures with their separate parameters $\theta_1, \theta_2, \ldots, \theta_k$. The unlabeled examples X_U are then fed to each of the k trained models to infer their corresponding probability vectors. We obtain $(f_{\theta_1}(x), f_{\theta_2}(x), \ldots, f_{\theta_k}(x))$ for each $x \in X_U$, and $f_{\theta_i}(x) \in \mathbb{R}^c \forall i$.

Sample Selection: Once the initial models on labelled data is trained, we use these models for pseudolabelling on the unlabelled data. For this step, we would ideally have those samples pseudolabelled, where the pseudolabel is more likely to be correct, as incorrect pseudolabels add label noise to the training data which may negatively affect convergence. We perform the following two steps for sample selection.

(1) So out of the $n - l$ unlabelled data samples, we calculate the entropy $H(x_i)$ of the prediction vector obtained from

the model M for each unlabelled sample x_i. The rationale for entropy is that having lower entropy prediction vector implies the model being more confident at those examples. In some SSL approaches (e.g. label propagation) use entropy, as a measure of uncertainty, to assign weight to the pseudo-labels. We also calculate $|L_1 - L_2|$ for every sample using the autoencoders that were trained. We sort them in increasing order of the confidence score $H(x_i) + |L_1 - L_2|$, as this score quantifies the confidence of M and the autoencoders A_1 and A_2 on the pseudolabels. We take the top k% samples from the sorted list for pseudolabelling.

(2) For all samples x_i that are selected from the above process, we find out whether the pseudolabel and the classification hint agree with each other. The subset of all samples selected by the previous step will be further selected and this is the set that would be used as pseudolabelled data in the next iteration of the self-training process.

Pseudo-labeling: For assigning pseudo-labels to the unlabeled examples, we take the *ensemble* of the predictions of the individual k models.

$$f_{ensmbl}(x) = \frac{1}{k}\sum_{j=1}^{k} f_{\theta_j}(x) \tag{4}$$

The unlabeled examples are then sorted in decreasing order of the confidence score $H(x_i) + |L_1 - L_2|$ of the ensemble prediction vectors $f_{ensmbl}(x)$, and first p examples with the lowest entropy are selected. We assign soft pseudo-labels $\hat{y}$ to these p examples as follows:

$$\hat{y}_i = f_{ensmbl}(x_i) \tag{5}$$

Guided Sharpening: Given the soft pseudo-label assigned to the unlabelled data which is the ensemble vector of the individual model predictions, we apply a sharpening function to reduce the entropy of the label distribution. However, we would like to have the strength of the sharpening process proportional to the confidence on the pseudolabel of that unlabelled sample. The sharpening function is controlled by a hyperparameter T, which is generally a heuristic for most applications. In our case we use the metric $|L_1 - L_2|$ as the hyperparameter for sharpening. It is given by

$$Sharpen(y, T)_i = \frac{y_i^{1/|L_1 - L_2|}}{\sum_{j=1}^{c} y_j^{1/|L_1 - L_2|}} \tag{6}$$

The soft pseudo-label is sharpened to obtain the final pseudo-label to be used for training the model.

$$\hat{y}_i^{sharp} = Sharpen(\hat{y}_i, |L_1 - L_2|) \tag{7}$$

Thus each unlabelled sample is sharpened depending on the confidence of its pseudolabel. the higher the confidence, the closer is the pseudolabel to a one-hot vector. This alleviates unnecessary label noise in the pseudolabels and helps the model focus more on those samples where the pasudolabels have a higher probability of being correct.

Training with unlabelled data: The enriched training data, containing both the original labelled samples as well as the pseudo-labelled examples, is now used to train next iteration classifier. Since the training data contains both hard (OHE) labels as well as soft (vector) labels, our loss function has two components - supervised loss $L_s(X_L, Y_L, \theta)$ using cross-entropy as defined in the Equation 1 for the hard-labelled examples and the unsupervised loss $L_u(X; \theta)$ using squared Euclidean distance as defined in the Equation 3 for the soft-labelled examples. The full model is trained using a loss which is the linear combination of these two losses i.e.

$$L_{SSL} = L_s + \lambda L_u \tag{8}$$

where λ is a hyperparameter to be chosen. If there is no pseudo-labelled data, the loss term reduces to the supervised loss only.

The procedure we follow is outlined in the following steps:

(1) Let $T_0 = (X^0, Y_0) = (X_L, Y_L)$ be the initial training data.
(2) Let X_U be the unlabelled data.
(3) for $i = 0$ to $i = N$, perform the following steps:
  (a) Train autoencoders A_1 and A_2 on fraud and non-fraud labelled data respectively.
  (b) Create k random samples of $T^{(i)}$ as described above.
  (c) Train separate models M_{θ_1}, M_{θ_2}....M_{θ_k} on each of the k samples.
  (d) For each $x \in X_U$, get its prediction vector from each of the k models. $(f_{\theta_1}(x), f_{\theta_2}(x), \ldots, f_{\theta_k}(x))$.
  (e) Compute the ensemble probability vector $f_{ensmbl}(x) \forall x \in X_U$.
  (f) Sort the unlabeled examples in terms of $H(x_i) + |L_1 - L_2|$, and choose first p examples $X_p \subset X_U$.
  (g) Select only those samples from $X_s \subset X_p$ that agree on the pseudolabel and classification hint obtained from A_1 and A_2.
  (h) Assign soft pseudo-labels $\hat{Y}_p = f_{ensmbl}(x)$ for all $x \in X_s$.
  (i) Sharpen the soft pseudo-labels to obtain $\hat{Y}_p^{sharp}$ for all $x \in X_s$ using $|L_1 - L_2|$ as the sharpening parameter.
  (j) Create the training data for next iteration as : $T^{(i+1)} = (X^{(i+1)}, Y^{(i+1)}) = (X^{(i)} \cup X_p, Y^{(i)} \cup \hat{Y}_p)$

4 EXPERIMENTS

In this section we outline the experimental evaluation that was carried out with the proposed method along with comparisons with related benchmarks. We begin the section with an illustration of the experiments, followed by other details. The experiments performed are as follows:

(1) **Learning using unlabelled data:** In this experiment we show the effectiveness of the proposed self-training algorithm on fraud detection. The first step is to learn a model on limited labelled data and then keep on adding pseudolabelled data from the unlabelled set, in an iterative fashion. For the elliptic dataset, we report the in the results of the algorithm after each iteration, on the other two databases we report the final results after completing all the iterations. The results are covered in Section 5.1.
(2) **Comparison with other approaches:** Fraud detection have primarily been performed with other approaches such as graph neural networks (and their variants), tree based models and so on. We show the comparison of the proposed self-training approach with these methods. The results are covered Section 5.2.

(3) **Ablation Studies:** We show several ablation experiments which help us tear apart each and every aspect of the proposed approach. The results are covered under Section 6.

4.1 Databases

We have utilized two different kinds of databases to evaluate our approach. We use the elliptic dataset to detect bitcoin transaction fraud [8] and Yelp reviews [6] and Amazon reviews [5] dataset to detect fraudulent online reviews. The databases are illustrated as follows:

4.1.1 Yelp Online Reviews: This database contains user reviews of hotels and restaurants. The total number of samples is 45,954 out of which around 14.5% are labelled as filtered (spam/fraud) and the rest are legitimate (non-fraud).

4.1.2 Amazon Online Reviews: This database also contains product reviews under the musical instruments category on Amazon's online marketplace. The total number of samples in this database is 11,944 out of which around 9.5% are fraudulent. This databases have been utilized previously to study the fraudster camouflage problem on which fraud detection can be evaluated.

4.1.3 Elliptic Bitcoin Fraud Dataset: This dataset contains bitcoin transactions and maps them to illicit (scams, malware, terrorist organizations, ransomware, Ponzi schemes, etc.) and licit (exchanges, wallet providers, miners, licit services, etc.) categories. The dataset is presented as a transaction graph, each node being a bitcoin transaction, and an edge represents the flow of bitcoins between one transaction and the other. The dataset contains 203,769 nodes/data samples, out of which 4,545 (around 2%) are labelled as illicit, 42,019 (around 21%) samples are labelled as licit and the rest are unlabelled.

4.2 Experimental Protocol

The semi supervised experiments have been performed using the proposed self training setup as explained in Section 2. For the Yelp and Amazon reviews dataset, the training-testing protocol is similar to [1]. For the Elliptic Bitcoin dataset. The protocol have been kept similar to [9], in order to compare with the results reported in that work.

In order to perform semi-supervised learning experiments, and at the same time enable us to compare our results with other reported results in the literature, we have used the same testing set across all our experiments in each database. During the training process, we use a part of the training data as labelled data and the rest as unlabelled data. This helps us evaluate the effectiveness of our approach of utilizing unlabelled data. The protocol is illustrated as follows:

Semi-Supervised Learning: For experiments in Semi-supervised learning for each of these (Tables 1, 2, and 3) we report the performance of the model trained only on labelled in the first row. The later iterations are performed by training of the combined set of labelled and some additional unlabelled data (proportion of licit and illicit samples is kept similar to the original positive class rate). In each iteration additional unlabelled data are introduced as pseudolabelled data, and are used with the labelled data pool.

Comparison with other approaches: Recently graph neural network (GNN) based methods have shown to be successful at fraud detection tasks. We show comparison with various SOTA GNN methods. The comparison with other approaches is separately performed for the online reviews datasets (Yelp and Amazon) on Table 4, and for the elliptic databases on Table 5. These comparisons are performed by keeping our protocol consistent with [1] for the online reviews datset and with [9] for the Elliptic Bitcoin Fraud dataset.

4.3 Evaluation Criteria

We evaluate the methods based on several criteria, such as accuracy, precision, recall, F1 score, Area under ROC (AUC) curve, Area under precision-recall curve (AUCPR) and micro averaged F1 score. Since fraud classification is a highly unbalanced problem (the fraction of fraud is very low compared to non-fraud), we rely more on F1 score and micro-averaged F1 score for evaluation.

5 EXPERIMENTAL RESULTS

In this section we review the experimental results on the three databases. We report the results under the following two subsections.

5.1 Semi-Supervised Learning

On the elliptic dataset we detect fraud in bitcoin transactions by training a model on limited labelled training data and improving this model by iteratively utilizing pseudolabelled data, which are unlabelled. Table 1 shows the results for each iteration of the proposed self-training approach. The first row in this table shows the baseline model which is trained on limited training data (around 30k data out of the 203k data available in this database). Thereafter in each row we show the amount of unlabelled data we added (in pseudolabelled form) and it can be easily observed that the F_1 score improves considerably. both precision and recall improves and the improvement in precision is noteworthy (from around 66% in the baseline model to 72% in the last iteration). This dataset is an ideal dataset for the proposed semi-supervised paradigm, since it has a large amount of unlabelled data.

On the Yelp and amazon reviews dataset we get similar gains from utilizing unlabelled data, although we had to simulate a semi-supervised learning protocol in those datasets, as they did not have any unlabelled data. In order to do this, we treated a part of the training data as unlabelled and pseudolabelled them in an iterative way. In each iteration the pseudolabelled data was used along with the labelled data for training. On the Yelp dataset, the F_1 score increases from 50.9% to 53.03% and on the Amazon reviews dataset the same is improved from 82.3% to 85.3%, while only using 40% of the training set as labelled data.

5.2 Comparison with other fraud detection approaches

We compare the results of proposed semi-supervised method with other recent graph based algorithms and also with simple models such as logistic regression and random forests. The comparisons are shown for the online reviews datasets (Yelp and Amazon) in and Elliptic Bitcoin Fraud dataset in Tables 4 and 5 respectively. For the Amazon and the Yelp online reviews dataset, we show

Table 1: Results for the proposed approach on the Elliptic dataset.

Experiment	Positive Pseudolabels	Labelled		Unlabelled		Precision	Recall	F1	Micro Avg F1	Acc	AUCPR	AUC
		licit	illicit	licit	illicit							
Labelled	-	26432	3462	-	-	0.75	0.60	0.66	0.96	0.96	0.62	0.91
iteration-1	13927	26432	3462	2550	1357	0.87	0.56	0.68	0.97	0.97	0.65	0.85
iteration-2	2331	26432	3462	5965	1550	0.85	0.61	0.71	0.97	0.97	0.68	0.91
iteration-3	5396	26432	3462	8241	3499	0.80	**0.67**	**0.73**	**0.97**	0.97	**0.73**	**0.91**

Table 2: Results for the proposed approach on the Yelp reviews dataset, by using only 40% of the labelled samples.

Experiment	Positive Pseudolabels	Labelled		Unlabelled		Precision	Recall	F1	Micro Avg F1	Acc	AUCPR	AUC
		licit	illicit	licit	illicit							
Labelled	-	11092	1774	-	-	0.872	0.324	0.472	0.893	0.893	0.7007	0.9059
Proposed	306	11092	1774	1729	233	0.844	**0.363**	**0.507**	**0.897**	**0.90**	0.6949	0.9026

Table 3: Results for the proposed approach on the Amazon reviews dataset, by using only 40% os the labelled samples.

Experiment	Positive Pseudolabels	Labelled		Unlabelled		Precision	Recall	F1	Micro Avg F1	Acc	AUCPR	AUC
		licit	illicit	licit	illicit							
Labelled	-	3112	231	-	-	0.8772	0.7958	0.8345	0.9786	0.9786	0.83801	0.9553
Proposed	96	3112	231	1236	5	**0.9382**	0.7835	**0.8539**	**0.9818**	0.9818	**0.8416**	**0.9585**

Table 4: Detection performance (%) (Area under ROC curve) on different proportions of labelled train data, on the Yelp and Amazon review databases. The rest of the training data have been utilized as pseudolabelled data, the ground truth labels of which were never made available during training.

Dataset	Train (%)	GCN	GAT	RGCN	Graph-Sage	Genei-path	Player-2vec	Semi-GNN	Graph-Consis	Care-Att	Care-W eight	Care-Mean	Care-GNN	Proposed
Yelp	10	50.94	55.45	55.12	54.20	56.29	50.15	51.68	62.07	70.21	71.02	71.85	73.31	**85.41**
	20	53.15	57.69	55.05	56.12	57.32	51.56	51.55	62.31	73.26	74.32	73.32	74.45	**87.46**
	40	52.47	56.24	53.38	54.00	55.91	53.65	51.58	62.07	74.98	74.42	74.77	75.7	**90.26**
Amazon	10	75.25	74.55	74.13	73.97	72.23	75.73	76.21	85.29	89.58	89.37	89.43	89.44	**95.30**
	20	75.13	72.1	75.58	73.97	71.89	74.55	73.98	85.5	89.58	89.68	89.34	89.45	**94.57**
	40	74.34	75.16	74.68	75.27	72.65	56.94	70.35	85.5	89.7	89.69	89.52	89.73	**96.41**

comparison with several graph based approaches, uysing different proportions of training data. On both the online reviews databases, the Care-GNN [1] method gave the previous state-of-the-art result, which the proposed semi-supervised algorithm outperforms with significant margin on all the three training data proportions. On the Elliptic Bitcoin Fraud dataset, we show comparison (Table 5) with other shallow classification models along with graph based methods namely GCN and Skip-GNN.

6 ANALYSIS OF RESULTS

In this section we analyze several aspects of the proposed self-training method and look into the different modifications that can be done to the proposed method.

6.1 Significance of proportion of pseudolabelled data classes

The amount of unlabelled data that we add in every iteration is chosen according to the decreasing entropy of the class-wise probability scores given by the models. In addition to this we can also observe that the models exhibit different performances, depending on what proportion of fraud and non-fraud data is added in each iteration. We perform an experiment (Table 6) where we only add illicit pseudolabelled data, and observe the the models perform

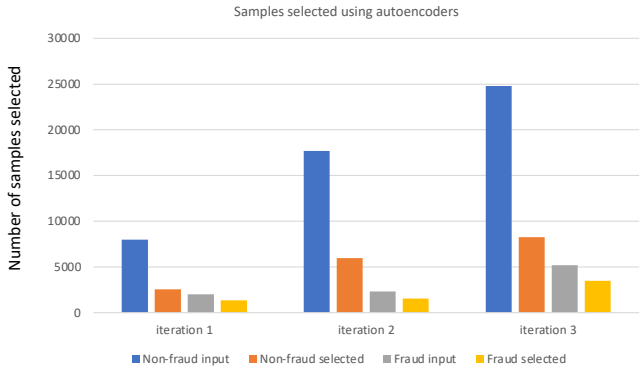

Figure 2: Histogram showing the number of samples selected using the classification hint from the autoencoders, from the Elliptic dataset.

poorly. This is in contrary to the general intuition that since it is a unbalanced classification problem, we should have got better performance by adding only the illicit class. The reason is that, since frauds are similar to anomalies, modelling anomalies may lead to poor performance from a classification model, which is exactly what happens in this case.

Table 5: Comparative results on the Elliptic Bitcoin Fraud database

	Logistic Regression	MLP	GCN	Skip-GCN	Proposed
Precision	40.4	69.4	81.2	81.2	**79.56**
Recall	59.3	61.7	51.2	62.3	**66.85**
F1	48.1	65.3	62.8	70.5	**72.65**
Micro-F1	93.1	96.2	96.1	96.6	**96.73**

Table 6: Experimental results on the Bitcoin Fraud dataset, while adding only the illicit pseudolabelled data during training.

Experiment	Positive Pseudolabels	Labelled		Unlabelled		Precision	Recall	F1	Accuracy	AUROC
		Licit	Illicit	Licit	Illicit					
Labelled	-	26432	3462	-	-	0.7453	0.5946	0.6615	0.9604	0.9056
iteration-1	13929	26432	3462	0	2000	0.3060	0.6685	0.4198	0.8799	0.8645
iteration-2	77925	26432	3462	0	2500	0.27703	0.68605	0.3946	0.8632	0.8686
iteration-3	81482	26432	3462	0	3000	0.2624	0.7192	0.3845	0.8504	0.87301

Table 7: Experimental results on the Bitcoin Fraud dataset, where in each iteration 10k new samples were added as pseudolabelled data. Lambda is increased from 0 to 20 as the training progresses.

Experiment	Labelled		Unlabelled		Precision	Recall	F1	AUCPR	AUROC
	Licit	Illicit	Licit	Illicit					
Labelled	26432	3462	-	-	0.7453	0.5946	0.6615	0.6232	0.9056
iteration-1	26432	3462	2526	1327	0.7764	0.5835	0.6663	0.4687	0.8672
iteration-2	26432	3462	5861	1520	0.8344	0.59095	0.69189	0.6749	0.90585
iteration-3	26432	3462	8175	3377	0.8255	0.6334	0.7168	0.6812	0.8994
iteration-4	26432	3462	11526	3178	0.8464	0.6057	0.7061	0.6869	0.89502

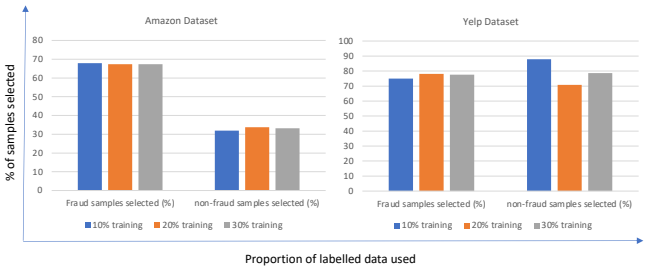

Figure 3: Histogram showing the percentage of samples selected using the classification hint from the autoencoders.

6.2 Gradually increasing lambda while training

Table 7 shows the results for semi-supervised learning experiments on the Elliptic database, where the value of lambda has been kept to low in initial iterations and then increased slowly in each iteration of self training. It can be seen that compared to the previous experiment on the same dataset (Table 1) the results in this case (benefit from the unlabelled data in each iteration) are comparatively better. In the first iteration of the self-training process, precision is at around 70%, wheras in the experiment shown in Table 1, the same is reached after the 4^{th} iteration.

6.3 Contribution of the Autoencoders

A natural ablation for the proposed approach is to observe the behaviour and contribution of the autoencoders, which were learnt separately on the fraud and non-fraud labelled data. During the pseudolabelling phase, we first sort the unlabelled data samples using the entropy of the predictions with the supervised model and the confidence metric obtained using the autoencoders. Once we pick up the top 10% data samples from the sorted list, we use the classification hint from the autoencoders along with the model's classification output to further refine this list. Figure2 shows the number of samples selected/refined using this process. We can see that a substantial number of samples are removed in each step where the classification hint from A_1 and A_2 and M do not agree. Similarly in Fig3 we can observe that the selection is more strict for fraud samples than non-fraud samples on the Amazon dataset, while the same for the Yelp dataset does not necessarily hold. This shows that a significant number of samples are removed in each step which prevents label noise during pseudolabelling.

7 CONCLUSION

We propose an end to end semi-supervised method for the fraud classification and exhibit the efficacy of the proposed method for two different fraud detection applications. The proposed method offers a complimentary approach to this problem which has been heavily investigated using other machine learning techniques. The pair of autoencoders provide useful cues for pseudolabelling and helps refine the unlabelled samples which prevents label noise. The proposed method is analyzed with respect to the effectiveness in semi-supervised learning as well as with comparative angle with some of the other results available on the same experimental protocol. This work enhances the state-of-the-art on both the fraud detection problems and is expected to open up a new set of methods

primarily focusing at semi-supervised learning in fraud/anomaly detection.

REFERENCES

[1] Yingtong Dou, Zhiwei Liu, Li Sun, Yutong Deng, Hao Peng, and Philip S Yu. 2020. Enhancing graph neural network-based fraud detectors against camouflaged fraudsters. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 315–324.

[2] Yingtong Dou, Guixiang Ma, Philip S Yu, and Sihong Xie. 2020. Robust spammer detection by nash reinforcement learning. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 924–933.

[3] Parisa Kaghazgaran, Majid Alfifi, and James Caverlee. 2019. Wide-ranging review manipulation attacks: Model, empirical study, and countermeasures. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*. 981–990.

[4] Parisa Kaghazgaran, James Caverlee, and Anna Squicciarini. 2018. Combating crowdsourced review manipulators: A neighborhood-based approach. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*. 306–314.

[5] Julian John McAuley and Jure Leskovec. 2013. From amateurs to connoisseurs: modeling the evolution of user expertise through online reviews. In *Proceedings of the 22nd international conference on World Wide Web*. 897–908.

[6] Shebuti Rayana and Leman Akoglu. 2015. Collective opinion spam detection: Bridging review networks and metadata. In *Proceedings of the 21th acm sigkdd international conference on knowledge discovery and data mining*. 985–994.

[7] Weiwei Shi, Yihong Gong, Chris Ding, Zhiheng MaXiaoyu Tao, and Nanning Zheng. 2018. Transductive semi-supervised deep learning using min-max features. In *European Conference on Computer Vision (ECCV)*. 299–315.

[8] Mark Weber, Giacomo Domeniconi, Jie Chen, Daniel Karl I Weidele, Claudio Bellei, Tom Robinson, and Charles E Leiserson. 2019. Anti-money laundering in bitcoin: Experimenting with graph convolutional networks for financial forensics. *arXiv preprint arXiv:1908.02591* (2019).

[9] Mark Weber, Giacomo Domeniconi, Jie Chen, Daniel Karl I Weidele, Claudio Bellei, Tom Robinson, and Charles E Leiserson. 2019. Anti-money laundering in bitcoin: Experimenting with graph convolutional networks for financial forensics. *arXiv preprint arXiv:1908.02591* (2019).

[10] Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V Le. 2020. Self-training with noisy student improves imagenet classification. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 10687–10698.

[11] Haizhong Zheng, Minhui Xue, Hao Lu, Shuang Hao, Haojin Zhu, Xiaohui Liang, and Keith Ross. 2017. Smoke screener or straight shooter: Detecting elite sybil attacks in user-review social networks. *arXiv preprint arXiv:1709.06916* (2017).