Chapter 2

Literature Survey

2.1 Ensemble Learning

Ensemble learning (EL) is useful in real-world problems with broad-spectrum. EL was primarily proposed to reduce the variance in the decision making systems. In an ideal scenario, EL expects the base classifiers to be diverse and accurate. Diversity among the classifiers is of utmost importance because of the robustness provided by it. Also, if each base classifier is accurate on the part of the entire problem, the ensemble is likely to cover the whole problem spectrum. In the past, there has been some research on utilizing diversity and accuracy of individual classifiers to come up with the final decision [13]. The popular approaches based on EL are bagging, stacking, random subspace, random forest, etc. These approaches have been used to solve various real-world problems, but we claim that the learning model needs to be adaptive for an open-set problem.

In [14], the authors propose a soft voting ensemble technique to combine the predictions, which, similarly to our proposed work, considers the individual performances while assigning weights to the classifiers.

2.2 Incremental Learning

Incremental learning has benefited many applications that require learning in pieces. In [15], the authors propose a method for an incremental summary generation where the task is to update the current summary of documents after encountering a new document. Their proposed approach finds the most significant sentence in the document that can be

replaced with a sentence in the current summary making it more accurate and updated.

Most of the approaches proposed for incremental learning so far make use of ensemble learning, where a set of classifiers are learned from each chunk of training data. Learn++ [16] is a family of algorithms widely used for various applications involving incremental learning [16–18]. Learn++ algorithms address the issues like the stability-plasticity dilemma, catastrophic forgetting, concept drift etc. Learn++ was initially proposed for training neural network pattern classifiers incrementally. It generates weak hypotheses and combines them using a weighted majority voting scheme. Learn++.NSE [17] enhances the previous version by accommodating concept drift in non-stationary environments. Learn++.NSE-SMOTE [19] enhances the previous versions by overcoming the problem of class imbalance in the streaming data. Irrespective of its popularity and broad applicability, Learn++ family of incremental learning models do not consider the properties of data while generating the base learners.

There are some cost-sensitive learning approaches proposed in the past that assign unequal misclassification costs to different classes [20]. Incremental learning becomes useful as it keeps track of the concept drift and manages it in a way that it does not increase false negatives. Another approach for detecting concept drift for incremental learning in non-stationary environments is proposed in [21]. The authors claim that the early detection of concept drift may result in improved accuracy.

Recently, there has been some research on handling the class imbalance while learning incrementally [22, 23]. In [24], the proposed learning model scales up to a large number of classes while managing the data imbalance between previously observed and new classes. While learning from the newly added data, it is essential to update the hypothesis accordingly. In [25], the old and new learning models are consolidated via a double distillation training objective. An unlabelled auxiliary data is exploited to consolidate the two models. In [26], the authors propose a dynamically updated ensemble algorithm for dealing with class imbalance and concept drift.

AILearn performs clustering to generate an ensemble of base classifiers in each learning phase. We claim that the advantages of clustering are two-fold: we can have a disjoint ensemble of classifiers (a must-have property in ensemble learning), and the base classifiers grasp the features of the data, which is helpful while classifying similar but unknown test instances.

Most of the research in the field of spoof fingerprint detection does not consider it as an application of incremental learning where fingerprints with novel spoof materials are added to the model. Various single-phase spoof detectors have been proposed in the past [27–30]. The performance of single-phase spoof detectors degrades drastically when new fingerprints generated from novel spoof materials are introduced. We claim that the fingerprint liveness detection's incremental aspect must be studied so that the spoof detectors can efficiently accommodate new spoof materials. The recent research on spoof fingerprint detection shows the incremental behaviour by using the existing learning algorithms based on ensemble learning, such as Learn++.NC [4, 18], whereas some of the researchers find a way to modify SVMs to accommodate new knowledge to the model [5]. We claim that while learning the base classifiers of the ensemble, the properties intrinsic to the dataset must be considered to form clusters of instances. Therefore, we propose a new learning algorithm and use it for spoof fingerprint detection.

To the best of our knowledge, AILearn is the first incremental learning algorithm that considers adaptiveness towards the similarity present in the data to generate a diverse set of base classifiers and integrate the ensembles incrementally in subsequent learning phases.

2.3 Spoof Fingerprint Detection

The field of spoof fingerprint detection has evolved rapidly over recent years, and yet it demands more attention. Fingerprint recognition systems are vulnerable to Presentation Attacks (PAs) made by various materials. In an adversarial environment, the attacker keeps evolving and attempts to break the system using a fabrication material on which the spoof detector has not been trained yet or by using a new sensor. Therefore, spoof fingerprint detection is an open-set problem, where the learning model needs to be adaptive towards the changing environment.

In recent years, ensemble learning has been used extensively to come up with solutions for spoof detection. In [4, 6], the authors propose mechanisms based on ensemble learning for tackling presentation attacks on fingerprint recognition systems. The robustness provided by multiple classifier systems is helpful in many problems. Ensemble-based classifiers can benefit spoof detectors provided the base classifiers are diverse and accurate.

In some attempts, the researchers use a single classifier to train on fingerprint images

and classify the test images [5, 31–35]. Some single classifiers outperform the ensemble-based classifiers, but mostly they are useful when the problem is targeted as a closed set problem.

Recently, there have been some attempts by researchers to address the poor generalization abilities of spoof detectors. In [36], the authors consider the poor generalization ability of the spoof detectors and propose an end-to-end patch level network to address the issue. In [37] and [38], spoof fingerprint generalization approaches based on CNN architecture are proposed.

We claim that our proposed work is different from the works mentioned above in the way it generates the ensemble of base classifiers. It considers the properties of data to decide the number of base classifiers to be integrated into the ensemble. Therefore, the individual classifiers are diverse and accurate, which is an ideal scenario by definition of ensemble learning. We claim that this feature of the proposed model makes it suitable for open-set problems like spoof fingerprint detection.

The spoof detection or PAD mechanisms can be broadly classified as hardware-based and software-based approaches. Hardware-based methods require a specialized set of devices to identify living traits such as blood pressure, skin distortions, wet, dry or moist skin. The software-based methods require a learning model trained on a feature set extracted from input fingerprint images. The trained model is used to classify test fingerprint images as "Live" or "Spoof". The winner of LivDet 2015 competition [30] used LBP handcrafted features and obtained state-of-the-art performance for binary classification. Several methods have been proposed based only on handcrafted features [39, 40].

The comparison between handcrafted and deep features has its importance in dynamic object tracking in a video given its initial state [41]. In [41] it was observed that, unlike image classification, object tracking is not benefited by the deep learning-based architectures for feature extraction. The deep features provide good robustness, but the results suggest that the handcrafted features outperform deep features because of their accurate target localization.

Another example of the comparison between both types of features for image classification is conducted by [42] for predicting overall survival (OS) in patients with Glioblastoma Multiforme (GBM). The authors conducted a study on the comparison and observed that deep features extracted via transfer learning outperformed the handcrafted features

for the given task.

Similarly, the authors in [43] studied the behaviour of these features on medical image classification and proposed a fusion called a combined deep and handcrafted visual feature for the task, which is claimed to be outperforming the individual feature types. In [44], the authors propose a fusion of both types of features for Facial Expression Recognition (FER) and achieve better performance than any of the two types of features independently. They considered Bag-Of-Visual-Words (BOVW) for handcrafted features and CNN-based architectures for automatic features, along with the SVM model for classification. In [36], the authors used both types of features for fingerprint spoof detection using patch level attention.

We provide a comprehensive study on the comparison between handcrafted and deep features for fingerprint liveness detection under various real-world scenarios. To the best of our knowledge, this study is the first attempt to compare both types of features while considering spoof detection as an *open-set classification problem*.

2.4 Automatic Hate Speech Detection

The scarcity of datasets on hate speech detection makes it challenging to analyze the models proposed in the past. Currently, Waseem and Hovy [45] and SemEval2019 [46] are the most popular datasets for the task. As we shall see in Chapter 7, these datasets have certain limitations and may introduce bias in hate speech detectors' performance. Detecting hate speech in a text is not an easy task, as it depends on various factors. The primary issue is its subjectiveness. It is a complex phenomenon that affects differently to different people based on their nationality, religion, ethics or even language nuances [47] [48] [49].

In the past, the problem of hate speech detection has been tried to solve by using general mechanisms for text feature extraction. These mechanisms include the Bag-of-Words (BOW) representation, the Term Frequency- Inverse Document Frequency (TF-IDF), word embeddings [50], deep learning models [51], [52], etc.

BOW is a text representation that creates a corpus of words present in the training data. In most cases, the frequency of words available in this corpus is used as the features and acts as the classifier's input. BOW representation is limited in determining the

context, and if the context is changed or used differently, it might lead to misclassification. TF-IDF is a numeric statistic that measures each word's importance in the corpus by its frequency in the training data.

In [7], the authors used a Recurrent Neural Network to extract features from the tweets and later classified these tweets using Gradient Boosted Decision Trees (GBDT) [53]. Their architecture is constituted by an embedding layer followed by a Long Short Term Memory (LSTM) network [54], a fully connected layer with three neurons and finally, a softmax activation. The authors use this architecture only as a feature extractor to turn a tweet (which is a sequence of words, $t =< w_1, w_2, .., w_k >$) into a sequence of vectors $E_t =< e_{w_1}, e_{w_2}, .., e_{w_k} >$ using the embedding layer. These vectors are averaged and fed into a GBDT classifier as inputs, and the resulting class label is achieved. The authors claim to yield a 93% F1 score at their best, but as corrected by [55], the actual best performance is 82% F1 score.

In [8], the authors use Bidirectional LSTMs (BiLSTM) as the recurrent layer that processes the input in both directions. The rest of their architecture is similar to [7]. The authors report their best performance as 94.4% F1 score, but as corrected by [55], the actual performance is 84.7%. The major flaw in their experimental setting is the method used for oversampling the minority class. The authors performed oversampling over the whole dataset and later partitioned it into train and test sets, which introduced a performance bias. Authors in [55] considered an example of oversampling the minority class three times and then partitioning the whole dataset into $15 - 85\%$ test-train split and observed that there is 38% probability that a particular instance may simultaneously belong to both train and test sets, which eventually increases the model's performance.

The problem of automatic hate speech detection is relatively new as compared to other NLP tasks. The basis of this problem can be established by studying similar tasks such as sentiment analysis on social media which has been thoroughly explored by the researchers. In sentiment analysis, the task is to classify tweets based on their sentiments, i.e. positive, negative or neutral [56], [57], [58], [59], [60], [61], [62], [63], [64], [65].

In [57], the authors use two feature selection methods, ReliefF and Multi-Verse Optimizer (MVO), along with SVM classifier, to accomplish the task. In [59], the authors propose a model for predicting the early impact of scientific research articles based on the sentiments expressed about them on Twitter. This is relatively a new research topic, and

it can benefit the researchers by examining the impact of their published articles. The study has been conducted on more than one Million research articles which are significant in itself. The authors claim that an article with positive and neutral sentiments expressed about it has a substantial impact on the research community. We expect that early estimation of the impact can be helpful in hate speech detection, where we can predict the overall effect of hateful posts by examining their early trends.

The more advanced research in the field of sentiment analysis is being conducted on Aspect-Based Sentiment Analysis (ABSA) [58], which provides fine-grained sentiments based on specific aspects. Another advanced area in this field is satirical text detection, which may influence the polarity of the statement in a considerable manner. In [66], the authors propose a mechanism based on psycholinguistic features to distinguish between satirical and non-satirical texts.

The difference between hate speech and offensive language is highlighted in [67]. The authors claim that the lexical methods to identify hate speech yield low precision as they fail to distinguish between the two categories. It was observed that the Hatebase lexicon flags a significant fraction of the tweets in their dataset as 'hateful', but only a few of them were found to be hateful by human annotators.

Some of the research in hate speech detection is also focused on regional languages. In [48], a three-tier pipeline is proposed to employs profanity modelling, deep graph embeddings, and author profiling to retrieve instances of hate speech in Hindi-English code-switched language (Hinglish) on Twitter. Similarly, other researchers have covered more languages such as Danish [68], Italian and German [49].

The above literature survey summarizes the work done so far on automatic hate speech detection and the related fields. Unlike this line of work, we deal with accelerating the process of automatic hate speech detection while managing user bias, data imbalance and cross-dataset generalization.