INTERPRETABLE FAKE NEWS DETECTION

A Thesis

by

SHIVA KUMAR PENTYALA

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Chair of Committee,	Xia Hu
Committee Members,	James Caverlee
	Ruihong Huang
	Xiaoning Qian
Head of Department,	Scott Schaefer

December 2019

Major Subject: Computer Science

Copyright 2019 Shiva Kumar Pentyala

ABSTRACT

Fake news is one of the most serious challenges facing the news industry today, which could result in adverse impacts on our society. Recent progress of deep neural networks (DNNs) has shown some promising results in detecting fake news. However, a critical missing piece of such detection is the interpretability, i.e., why a particular piece of news is detected as fake. This thesis investigates several approaches for explainable detection of fake news, including its several forms: texts, images and videos. First, we study some techniques to efficiently explain the output prediction of any given news. It sheds light on the decision-making process of the detection models and could illustrate why the detection model succeeds or fails. Second, we show that refining those explanations can enhance the model's generalization ability. To make this refinement process feasible, we propose an active learning strategy to identify the challenging examples in the training data that are responsible for the model's overfitting. Several experiments have been conducted to demonstrate the effectiveness of our active learning strategy for image/video-based fake news detection. Third, we propose an interactive explainable detection system for language based (text) fake news to help end-users identify the news credibility. We provide several explanations like word/phrase importance, attribute importance, linguistic feature importance, and supporting examples, which could help end-users understand why the system makes that decision.

ACKNOWLEDGMENTS

The 2-year M.S. study at TAMU was an incredible experience. There are several people that I wish to acknowledge who not only made this thesis work well accomplished, but also brought unlimited amount of joys and encouragement to this amazing journey.

I would first like to thank my advisor, Dr. Xia Hu, for his enthusiasm, inspiration, and guide throughout my M.S. study. He opened the door of artificial intelligence for me at the very beginning and encouraged me to further my research at TAMU. During this study we collaborated for explainable artificial intelligence (XAI) project and i learnt a lot from him during this time. His sharp intuition and endless passion motivate me to dig deeper into the problems and discover something new. I couldn't achieve all of this without his support.

I would also like to thank my thesis committee, Dr. James Caverlee, Dr. Ruihong Huang and Dr. Xiaoning Qian, for their kind support and advice on my research.

I would like to thank our XAI project advisors, Dr. Shuiwang Ji and Dr. Eric Ragan for their valuable suggestions during the project meetings in the past two years. I feel the discussions with dear projectmates have better me alot as a researcher to think outside the box and also to keep track of research advances in this field. Projectmates includes Fan Yang, Mengnan Du, Sina Mohseni, Ninghao Liu, Yi Liu, and Hao Yuan. Also, Chapter 2 and 4 in this thesis have been collaborated closely with Mengnan Du. Mengnan's insights on developing interpretable deep neural networks give me a brand new and inspiring perspective on the problems I work on.

The DATA lab at TAMU got a friendly and inclusive environment. Attending research presentations given by labmates during weekly lab meetings help me see a broader exploration of deep learning research in various domains. These dear labmates at DATA lab inaddition to the above mentioned names made my graduate school journey more colorful and fulfilling: Qingquan Song, Haifeng Jin, Yuening Li, Diego Martinez, Xiao Huang, Daochen Zha, Kwei-Herng Lai, Yi-Wei Chen, Anurag Kapale, Kaixiong Zhou.

CONTRIBUTORS AND FUNDING SOURCES

Contributors

This work was supervised by a thesis committee consisting of Dr. Xia Hu, Dr. James Caverlee and Dr. Ruihong Huang of the Department of Computer Science and Engineering and Dr. Xiaoning Qian of the Department of Electrical and Computer Engineering. All work for the thesis was completed independently by the student.

Funding Sources

Graduate study was supported by a ISFAA grant from Texas A&M University.

NOMENCLATURE

DNN	Deep Neural Network
RNN	Recurrent Neural Network
CNN	Convolutional Neural Network
CAM	Class Activation Map

TABLE OF CONTENTS

Page

LIST OF FIGURES

FIGURE Page

LIST OF TABLES

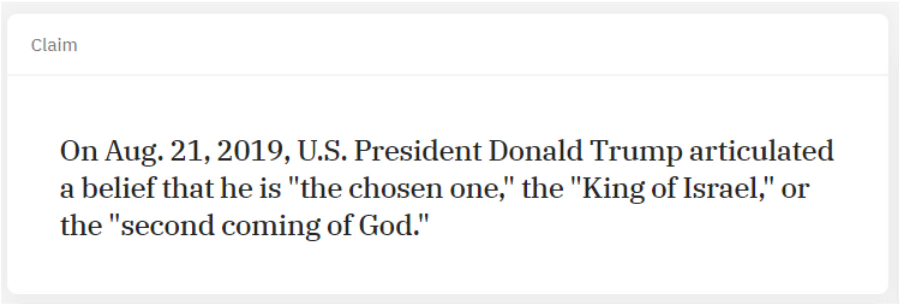
1. INTRODUCTION AND LITERATURE REVIEW

"Fake News" comes in multiple forms (articles, images, videos) and multiple flavors (misinformation, disinformation, propaganda, satire, rumors, hoaxes, click-bait and junk news). In general, fake news is defined as the news with intentionally false information [6, 7]. Some studies have found that such news arise at the intersection of busy social networks and limited attention spans. In a perfect world, carefully reported and factually accurate news would go viral. But that isn't necessarily the case. We keep falling for fake news. Recently, Elon Musk's April Fool tweet saying "Tesla goes bankrupt" made their stock fell 5% the following day [8]. There were times when it fell by 7% because of his tweet on a normal day [9]. Also, the reach of fake news during the 2016 U.S. presidential election campaign for top-20 fake news pieces was, ironically, larger than the top-20 most-discussed true stories [10]. While "fake news" may be a buzzword, it's certainly no joke. Wide spread of fake news can cause serious negative impact on our society and thus it become critically important to be able to curtail the spread of fake news on social media, promoting trust in the entire news ecosystem. Inorder to do that first we need to understand it's flavours, challenges, effects etc.

1.1 Forms and Flavors

Generally, fake news come in either textual or vision form as shown in Fig. 1.1. In the past, textual form is more commonly seen (ex: spam emails, reviews etc.). However, with the advancements in artifical intelligence, even vision based fake news got super realistic such as, the difference between true and fake images are so subtle, even human eyes are hard to distinguish them. For instance, Fig. 1.1c shows a fake image of obama created by a GAN-based technology called DeepFake [11]. Also in 2018, a video released by BuzzFeed [12] making Barack Obama voice his opinion on Black Panther (“Killmonger was right”) and call President Donald Trump “a total and complete dipshit.” [13] attracted huge attention of even research communities.

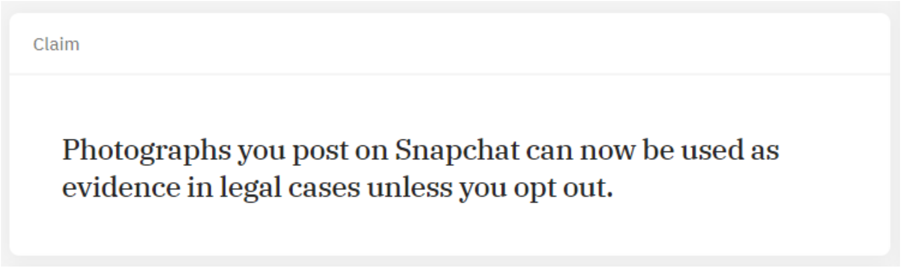
Here, we will review several flavors of fake news that can be found on the web. All of them



Claim

On Aug. 21, 2019, U.S. President Donald Trump articulated a belief that he is "the chosen one," the "King of Israel," or the "second coming of God."

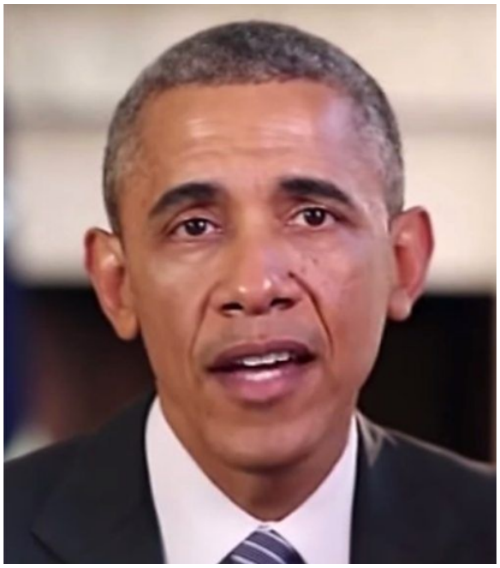
(a) Fake article on the web about politics [14].



Claim

Photographs you post on Snapchat can now be used as evidence in legal cases unless you opt out.

(b) Fake article on web about social media [15].



(c) Face manipulation by DeepFake [11, 16].

Figure 1.1: Example web news analyzed by a fact-checking website, Snopes [1].

satisfy the following definition for fake news: a claim which did not originate from news events and has not been verified while it spreads from one person to another[17, 7, 6]. *Misinformation* is defined as the inaccurate or misleading info [18]. It can spread unintentionally [19] due to honest reporting mistakes or due to incorrect interpretations [20]. *Disinformation* works in contrast to Misinformation. Defined as false information that is spread deliberately to deceive people [18] or promote biased agenda [21]. *Hoaxes* are defined as humorous and mischievous [22] that are similar to disinformation in terms of intentionally conceiving to deceive readers [22]. *Satire* whose primary purpose is to criticize or entertain the readers. They are characterized by irony, humor, absurdity and they can mimic true news [23]. However, similar to hoaxes, they could be harmful when shared out of context [24, 25]. *Propaganda* puts information which tries to influence the opinions, emotions, and actions of target audiences by means of deceptive, selectively omitted and one-sided messages. Typically, purpose of this category is political, ideological or religious [26, 21]. *Click-bait* is a low quality journalism that are intended to attract traffic and monetize via advertising revenue [21]. *Junk news* is more generic and it aggregates different types of information, from propaganda to hyper-partisan or conspiratorial news and information. Typically, it refers to the overall content that pertains to a publisher rather than to a single article [27].

1.2 Effects on Society

With rapid usage of social media, fake news can cause much diverse effects on the society. Three main effects include:

- Could change the way people respond to legitimate news [28].
- Significantly weakens the public trust in governments and journalism [10].
- Rampant ’online’ fake news can lead to ’offline’ societal events [29].

In general, below four domains accounts for most of the attention on fake news. For each domain, we will provide the striking effects that the world has recently experienced, that indeed responsive for today’s explosive growth of attention on fake news.

- **Politics** Accounts for most of the attention as highlighted in [30]. Some striking effects include: The US presidential elections in 2016 have officially popularized the term ’fake news’ to the degree that it has been suggested that Donald Trump may not have been elected president [6]. Likewise, 2016 UK Brexit referendum [31] and the 2017 France presidential elections have been impacted by fake news [32].

- **Finance** Crisis caused by a false tweet concerning president Obama was injured in an explosion. This tweet wiped out $130 billion in stock value [33, 29].

- **Crime** As a consequence of the Pizzagate fake news, shootout occurred in a restaurant [7]. Over 1 million tweets were related to the fake news story “Pizzagate” by the end of 2016 presidential election [34].

- **Health** Diffused mistrust towards vaccines during Ebola and Zika epidemics [35].

1.3 Challenges in Detection

We present here few unique challenges involved in detecting fake news on social media.

- **Just content doesn’t help.** Fake news are intentionally written to deceive the readers and to mimic traditional news outlets, resulting in an adversarial scenario where it is very hard to distinguish true news from false ones simply based on their content [7, 36]. For example, image in Fig. 1.1c looks super realistic and without additional information it’s tough to identify it as fake. Same with language based (text) fake news as shown in Fig. 1.1.

- **Manual fact-checking doesn’t scale.** Rate and volumes at which fake news are produced overturn the possibility to fact-check and verify all items in a rigorous way, i.e. by sending articles to human experts for verification [36]. Also, there has been rise of fact-checking websites, such as Snopes.com and PolitiFact.com, where people research claims, manually assess their credibility, and present their verdict along with evidence (ex., background articles, quotations, etc.). However, this manual verification is time-consuming and not feasible specially for large volumes.

- **Limited labeled data.** Social media platforms impose limitations [37] on the collection of public data and as of today the community has produced very limited quality datasets. Also, it's much difficult to find annotators with knowledge about particular news creator, news subject etc. for fake news flavors like opinion based contents, and humorous stories like satires.

- **Feature engineering difficulties.** It's also difficult to develop machine learning models with substantial feature modeling and rich lexicons to detect bias and subjectivity in the language style. On the other hand, it's very likely for deep learning models to get overfitted to small datasets.

1.4 Simplifying the Definition

The definition of fake news used by each author is important to us, as the term became very diffused by researchers, journalists, politicians, and users throughout the media. The fake news can have several flavors as seen in section 1.1. Although, all of those flavors have exclusive attributes that separate them in their respective meaning group, but all converge to the same semantic meaning, that is of an information that is unverified, of easy spread throughout the net, with the intention of either block the knowledge construction (by spreading irrelevant or wrong information due to lack of knowledge of the theme) or either manipulate the readers opinion [38, 39, 40].

Due to very broad definition of fake news (for example opinion based contents, and humorous stories like satires), it's difficult to get unbiased (labeling) datasets or to create generalizable machine learning solutions. We need to restrict the definition, not only for conceptual enlightenment, but for assertiveness in our revision, and meta-modeling reference of future works, as this would be the foundation for experiments.

In this work we will restrict the definition of fake news to the one used in [7], which is "a news that is intentionally and verifiable false". Note that this definition shares similarities to our initial definition "news with the intent to deceive and false factual content, or the news with intentionally false information". However our new definition is simplistic, since it does not cover half truths, opinion based contents, and humorous stories, like satires. Specifically for vision based fake news, we will use datasets with modified images and their corresponding true images.

1.5 Current Detection Methods

To help mitigate their adverse effects, it is essential that we develop methods to detect the manipulated forgeries. Recent advances in deep learning, overcome some of the challenges in section 1.3 by incorporating news context inaddition to the news content. Context can be user profile info [41, 42], post-based info like users social responses/comments [43, 44], network info [45, 46, 47]. However, most of the vision based fake news just use news content (pixels) information because of lack of the datasets with additional context information. So detection methods vary based on the modality of fake news (vision/text). We have a brief survey on several lines of work in vision and language that are related to this topic.

1.5.1 Vision

In this section, we briefly review three lines of research that model vision based fake news detection. Current developments in forgery detection field mostly formulate it into a binary classification problem, roughly falling into two branches: CNN based approaches and artifacts based methods. The first category takes either the whole or partial image as input and then classify it as fake or not by designing diverse architectures of convolutional networks [48, 49, 50]. While the second categories relies on hypothesis on artifacts or inconsistencies of a video or image, such as lack of realistic eye blinking [51], face warping artifacts [52], and lacking self-consistency [53]. mismatched color profiles [54]. However, both these two categories of methods tend to overfit to the data in the training set and perform poorly on new unseen manipulations [55]. Considering the two distinct characteristics which differentiate forgery detection with typical image classification task, in this work we aim at designing relevant models to enhance generalization performance of the forgery detection problem.

Although autoencoder-based structure has been demonstrated to be successful in many image outlier detection problems [56], using autoencoder for forgery detection still is a challenging problem due to two main reasons. Firstly, it is a fine-grained classification task. The difference between true and fake images are so subtle, even human eyes are hard to distinguish them. Secondly, the forgery region only occupies a small ratio of the whole image. If no explicit supervision imposed to the learning process, the model may fail to focus on the forgery region. Instead, they may concentrate on non-forgery part, and learn spurious correlations to separate true and fake ones. This would significantly hinder their generalization ability. So there hasn't been much research on using autoencoder based approaches for this task. However, recently [55] uses an autoencoder based approach for fake image detection but their generalization accuracy is still around 50% on all their datasets.

1.5.2 Language

Language based (text) fake news detection has been traditionally formulated as a supervised binary classification with a detection methods, from traditional machine learning (Logistic Regression, Support Vector Machines, Random Forest) to deep learning (Convolutional and Recurrent Neural Networks) and to other models (Matrix Factorization, Bayesian Inference). The main challenge here is to get a labeled dataset with good quality. Current datasets typically fall into one of the three categories - content based (short claims, twitter post etc.), context based (diffusion networks, users' profile, metadata), both (twitter post with its user profile info, facebook post with comments etc.) [7, 57, 58]. So, we will sequentially review the methods by starting from those contributions which focus only on content-based features; only the context and finally those that consider both aspects.

- **Content-based detection.** Here we review the detection methods that solely analyze the textual content of news, e.g. body, title. Decisions made just based on textual content are likely to capture specific writing styles [59] and sensational emotions [60] that frequently occur in fake news contents. In general, several machine learning approaches [61, 62, 63] have been explored using lexical or special linguistic features like ngrams, LIWC [63], punctuation, syntax and readability. Also in later years, several supervised deep learning approaches

[64, 65, 66, 67, 68, 69] have been explored that showed much better results compared to machine learning methods. Recently, there was also an unsupervised approach [70] to distinguish different categories of fake news (from satire to junk news), based only on the news content. Their method involves tensor decomposition of documents which aims to capture latent relationships between articles and terms and the spatial/contextual relations between terms. Further they use an ensemble method to leverage multiple decompositions inorder to discover classes with lower outlier diversity and higher homogeneity. Their experimental results outperform other state-of-the-art clustering techniques in correctly identify categories of fake news. However, in general modeling a detection method just based on news content is less likely to have good generalization ability or to handle real-world data efficiently.

- **Context-based detection.** Here we review the research contributions which are (social) context-based in the sense that they utilize information derived from social interactions between users while making a decision. Some examples of such interactions include likes, comment and (re)tweets, user connections etc. These interactions can also be grouped into user-based, post-based and network-based. Accordingly, there are several works that learn these user-based features from user profiles [41, 42], post-based features from users social responses (in terms of stance) [71], topics [43] and credibility [44], learn network-based features by constructing either diffusion networks [72] or propagation networks [45, 46] or interaction networks [47]. For example, features learnt with the help of propagation networks concentrate on propagation of messages carrying malicious items in social networks [72]. Propagation of news items is also taken into account by [73] which basically combines convolutional and Gated Recurrent Units (GRU) [74] to model diffusion pathways as multivariate time series, where each point corresponds to the characteristics of the user retweeting the news. Both these [73, 72] sound promising because of their analysis on user profiles and online news sharing cascades. Despite of the inherent complexity of both techniques (also limited datasets employed), it looks like network-based approach focusing on social responses might effectively detect deceptive information. On the other hand, first unsupervised approach to false news detection is provided by [75], where veracity of the news and users credibility are treated as latent random variables in a Bayesian model, and the inference problem is solved using collapsed Gibbs sampling approach [76].

- **Content and Context based detection.** Here we will review the research contributions that consider both news content and the associated (social) context interactions in making the decision. This work [77] uses a deep learning approach to analyse user behaviours in terms of lag, activity and shown that the source users who promote the news is a promising feature for the detection. Later [26] infers different deceptive strategies (misleading, falsification) and different types of deceptive news (propaganda, disinformation, hoaxes). Interesting part about their work is that beside traditional content-based features (syntax and style), they employ psycho-linguistic signals like biased language markers, moral foundations and connotations. In addition, they also inspect social responses on Twitter as to infer different deceptive strategies and types of malicious information. Recently [47] proposes an approach that employs tri-relationship among publishers, news items and users. Their results show that the social context could be effectively exploited inorder to improve fake news detection.

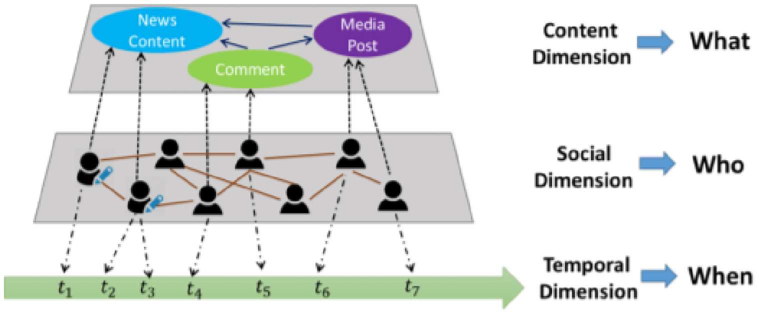


Figure 1.2: Example of content and context based detection. This also considers temporal dimension for better understanding of news behaviour over time [2].

1.6 Need for Interpretable Detection

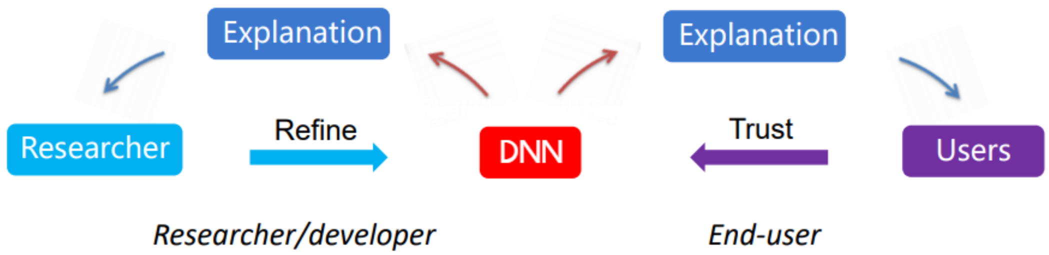


Figure 1.3: Need for explaining a deep neural network prediction from two perspectives: End-user, Researcher/developer.

There is a growing interest among the academic and industrial community in interpreting deep learning models and gaining insights into their working mechanisms. In our case, being able to interpret/explain why a news was determined as fake or true is much desirable for several reasons illustrated below.

1.6.1 End-user Perspective

For end-users, explanation will increase their trust and encourage them to adopt deep learning systems. With explanation, the area experts could provide realistic feedbacks. Eventually, new science and new knowledge which are originally hidden in the data could be extracted. Also, explanation can further motivate the users to provide more/better annotations.

1.6.2 Researcher Perspective

From the perspective of deep learning system developers/researchers, the provided explanation can help them better understand the problem, the data and why a model might fail, and eventually help in increasing the system safety. Typically, this perspective has below two applications.

- **Model Validation.** Interpretations could help to examine whether a deep learning model has employed the true evidences instead of biases which widely exist among training data.

Also, deep learning models may rely on gender, topic and ethnic biases to make decisions. Interpretability could be exploited to identify whether models have utilized these biases to ensure our models don't violate ethical and legal requirements.

- **Model Debugging.** Explanations could be employed to debug and analyze the misbehavior of models when they give unexpected or wrong predictions. A representative example can be adversarial learning [78]. Recent work demonstrated that deep neural networks can be guided into making erroneous predictions with high confidence when processing deliberately or accidentally crafted inputs [78, 79]. However, these inputs are quite easy to be recognized by humans. So in such cases, explanation facilitates researchers to identify the possible model deficiencies and analyze why these models may fail.

1.7 Current Interpretability Methods

Most of the DNN interpretation techniques can generally be grouped into two categories as shown in Fig. 1.4 : intrinsic interpretability and post-hoc interpretability, depending on the time when the interpretation is obtained. Intrinsic interpretability is achieved by constructing self-explanatory models that incorporate interpretability directly to their structures. The family of this category includes attention model etc. In contrast, the post-hoc one requires creating a second model or even a simple heuristic to provide explanations for an existing model. The main difference between these two groups lies in the trade-off between model accuracy and explanation fidelity. Inherently interpretable DNN's could provide accurate and undistorted explanation but could sacrifice prediction performance to some extent. The post-hoc type way is limited in their approximate nature while keeping the underlying DNN accuracy intact.

1.7.1 Vision

As CNN is the most dominant architecture used in vision community, we will mainly focus on explanation methods developed for CNN-based detection networks. The detection models needs to possess local interpretability, which could indicate which region is attended by the model to make its decisions. The benefit is that explanations can help users better understand their models or in

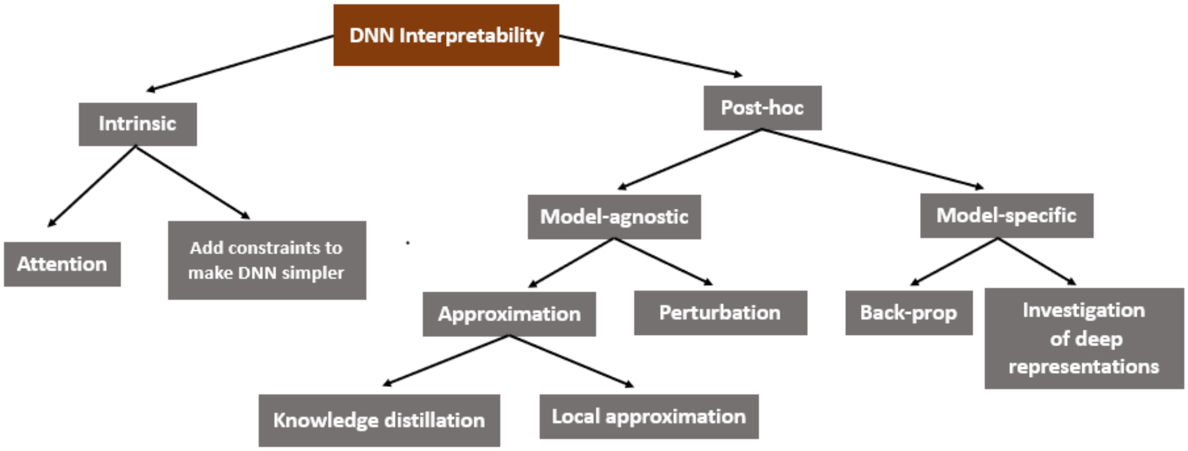


Figure 1.4: Techniques to interpret a Deep Neural Network (DNN) [3, 4].

gaining trust about their model decisions. This can further motivate them to improve/provide more annotations for some domain of training data on which model has bad performance.

- **CNN Global Interpretation** The global interpretation enables users to understand how the CNNs work globally by inspecting the representations captured by the neurons at different intermediate layers of CNNs [3]. Among different strategies to understand CNN representations, the most effective and widely utilized one is through finding the preferred inputs for neurons at a specific layer. This is generally formulated in the activation maximization (AM) framework [80]. This framework ultimately generates a visualization that could tell what individual neuron is looking for in its receptive field. This method can be used for arbitrary neurons, ranging from neurons at the first layer to the output neurons at the last layer, to better understand what is encoded as representations at different layers.

- **CNN Local Interpretation** Local explanations target to identify the contributions of each feature in the input towards a specific prediction of the deep neural network. These local interpretation methods can be further classified into the following three main categories: Back-propagation, Perturbation, and Investigation of representations in intermediate layers.

  Back-propagation based methods [81, 82, 83] calculate the gradient or its variants, of a par-

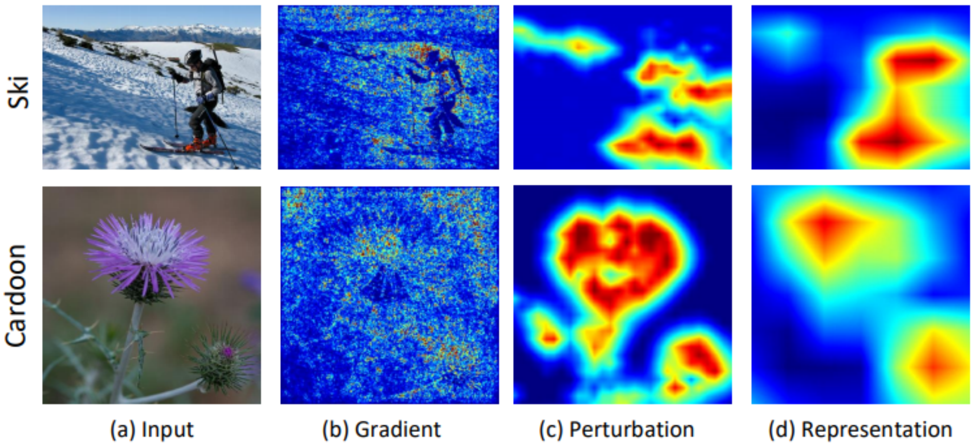


Figure 1.5: CNN local interpretation heatmaps produced by (b) Back-propagation, (c) Perturbation, (d) Investigation of representations.

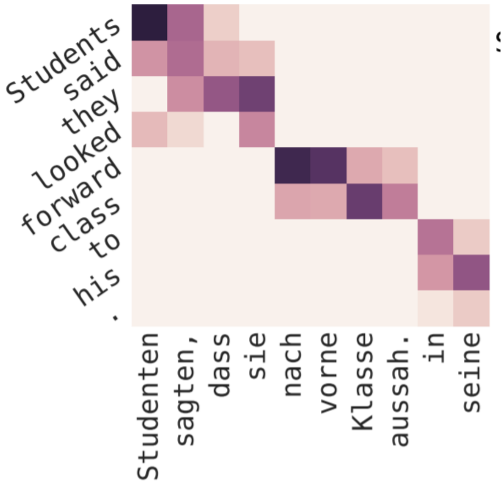
ticular output with respect to the input to derive the individual pixels in the input image. However, this method is limited in its heuristic nature and may generate low quality explanations that are noisy, as shown in Fig. 1.5(b).

Perturbation based methods [84, 85] tries to answer the question: which parts of the input, if were not seen by the model, would change its prediction the most? This can also be framed as how prediction score changes when few input features are altered? The perturbation is performed sequentially across features which in our case are pixels, to determine their contributions, and can be implemented either with omission or occlusion. For omission, a feature is directly removed from the input and for occlusion, the feature is replaced with a reference value, such as gray value of pixel or mean of the input pixel values. However, occlusion raises an additional concern that new evidence may be introduced and that can be used by the model as a side effect [84]. Thus we should be cautious when selecting reference values inorder to avoid introducing extra pieces of evidence. If you notice, pixel-wise perturbations could be computationally very expensive because of high dimensional inputs, since pixels need to be perturbed sequentially. To overcome this complexity, perturbation can be done at superpixel level with the help of a mask followed by gradient descent optimization. One recent work [85] uses an optimization framework to learn a perturbation mask, which explicitly preserves the contribution values of each feature. These superpixel level explanations would be more meaningful compared to pixel-level explanation as shown in Fig. 1.5(c). Although superpixel level methods has drastically boosted the efficiency, generating an explanation still requires several forward and backward operations.

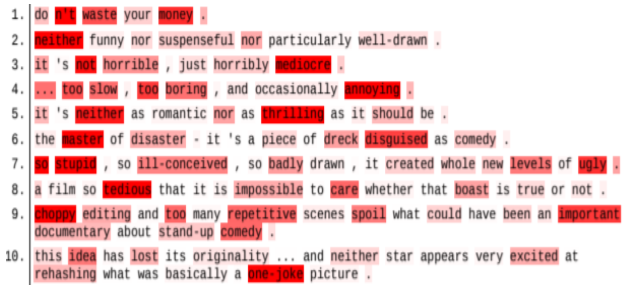
Investigation of representation based methods [86, 87, 88, 89] explicitly utilize the deep representations of the input to generate heatmaps(or saliency maps). Either perturbation or back-propagation based explanations ignore these intermediate layers of the DNN which are likely to contain rich semantic information. Examples of this kind of interpretation methods are CAM [88], Grad-CAM [89] which generate saliency maps by combining the feature maps (or channels) in the intermediate CNN layers heuristically. The difference between CAM and Grad-CAM is that the former can only be applied to a small set of CNN classifiers with global average pooling layer prior to the output layer, while the latter has no such requirement as it combines the intermediate feature maps using gradient, and thus can be applied to a wider range of CNN architectures. However, one main advantage of the CAM explanation method is that it is end-to-end differentiable, amenable for training with back-propagation and updating CNN parameters. In general, people use the last convolution layer feature maps to get meaningful CAM or Grad-CAM saliency maps, as last convolution layer is known to learn more abstract representations with high levels of semantics.

1.7.2 Language

All explanation methods can be generally grouped into two categories: intrinsic and post-hoc, depending on the time when the interpretability is obtained [90]. As RNN is the dominant architecture used by NLP community, we will mainly review explanation methods from those two categories that could provide interpretations for RNN predictions. These methods can be further grouped into two: model-agnostic and model-specific. For model-agnostic methods, we regard the model to be explained as black box by which we only have access to its input and output. On the other hand, model-specific methods need to know model architecture and parameters.



(a) Attention based explanation for NMT task [91]



1. do n't waste your money .
2. neither funny nor suspenseful nor particularly well-drawn .
3. it 's not horrible , just horribly mediocre .
4. ... too slow , too boring , and occasionally annoying .
5. it 's neither as romantic nor as thrilling as it should be .
6. the master of disaster - it 's a piece of dreck disguised as comedy .
7. so stupid , so ill-conceived , so badly drawn , it created whole new levels of ugly .
8. a film so tedious that it is impossible to care whether that boast is true or not .
9. choppy editing and too many repetitive scenes spoil what could have been an important documentary about stand-up comedy .
10. this idea has lost its originality ... and neither star appears very excited at rehashing what was basically a one-joke picture .

(b) Back-propagation based explanation for text classification [92]

Figure 1.6: Example explanations in NLP.

- **Model-specific explanation.** Typically, this type of explantion either take advantage of attention layers in the model or gradients of the parameters involved inbetween class of interest and the model input. *Attention* based methods [93, 94, 91, 95] which are widely utilized to explain predictions made by Recurrent Neural Networks (RNNs). This method gives users the ability to interpret which parts/words of the input are attended by the model for it's prediction. Also, this approach has been used in multi-modal tasks like image caption generation [94] where a convolution neural net is employed for encoding image into a vector which is later used by RNN with attention mechanisms to generate corresponding descriptions about the encoded image. In this process, during each word generation, model changes its attention to reflect the relevant parts of the image. Visualizing the attention matrix for individual predictions could tell us what the model is looking at when generating a word. Another dominant task which uses attention mechanism is machine translation [93]. At decoding stage, the neural attention module assigns different weights to the hidden states of the decoder, which allows the decoder to selectively focus on different parts of the input sentence at each step of the output generation. Through visualizing the attention scores, users could understand how words in one language depend on words in another language for correct translation as shown in Fig. 1.6a. *Back-propagation* based methods [92, 82, 96, 97, 81] computes the gradient or its variants of the model output with respect to the input, for a specific class of interest. This is typically done using traditional back-propagation to identify the words whose variation would lead to the significant change of output probability. It is tricky to say this method belongs to model-specific category and it may change depending on the way we define model-agnostic/model-specific. As our definition for model-specific explanation involves utilizing model parameters, thus it falls here. This approach is first proposed in vision for image classification [80], later adapted to language tasks like text classification etc. However, unlike in vision tasks there is no unique way of using this explanation method for language tasks because of word embedding layer. So there has been different variants of this method such as computing gradients with respect to individual entries in word embedding vectors, and then the L2 norm [97] or the dot product of the gradient and the word embedding [96] inorder to reduce the gradient vector to a scalar, representing the contribution of a single word. Also, there are some works that propose to back-propagate different signals to the input, such as the relevance of the final prediction score through each layer of the network onto the input layer [92, 82], or only propagating positive gradient signals in the back-propagation process [81]. However, the pitfall of this explanation method is that the heatmaps would be bit noisy and we need to employ some postprocessing to make explanations look good. Some examples for this type are shown in Fig. 1.6b. *Decomposition* based methods [98, 99, 87] tries to utilize the deep representations of the input to explain the DNN prediction. Above back-propagation based method, ignore the intermediate layers of the DNN that might contain rich information for interpretation. By modeling the information flowing process of the hidden representations in LSTM models, the LSTM prediction is decomposed into additive contribution of each word in the input sentence [98, 99]. There has been a recent work which extends this idea to all RNN architectures [87] by also enabling the flexibility to generate word/phrase/clause level explanations. The decomposition result can quantify the contribution of each individual word to our RNN prediction. Unlike approximation based methods, these explanations would be more meaningful/faithful to the original model's decision making process compared to the explanations taken from it's approximated model version. This is because deep representations serve as a strong regularizer, increasing the possibility that the explanations faithfully characterize the behaviors of complex model, thereby reducing the risks of generating surprising explanations.

- **Model-agnostic explanation.** *Perturbation* based methods [100, 101, 102] in NLP try answers the question: which words in the input, if were not seen by the model, would change its prediction significantly? This can also be framed as how prediction score changes when few input words or word features were altered? The motivation of this method is that if the most important word for a prediction is perturbed, then it will cause the largest probability drop of the output for the target class. Perturbation can be induced in two ways: occlusion [102] and omission [100, 101]. For occlusion, word is replaced with a baseline input where a zero-valued word embedding is utilized as replacement [102]. While for omission, we directly delete the word [100]. This approach is easy to implement but the problem is, it cannot guarantee meaningful explanations because of the word order modified by omission or occlusion. Both of them could make the sentence nonsensical. Since word order is an essential factor for RNNs, these word order distortions may trigger the adversarial side of RNN resulting in unfaithful explanations. *Approximation* based methods [103, 104, 105, 106, 107, 108] try to approximate original deep-learning/black-box model with a shallow/white-box model. This can be done either with model extraction based methods like mimic learning [108, 107, 109, 105, 106]. or with local approximation based methods like LIME [103] etc. *Mimic learning* is a model extraction based method proposed in [107, 105, 108], which tries to transfer the knowledge of a pre-trained complex/deep model (called teacher) to a simple/interpretable model (called student) without sacrificing much on accuracy. One way of doing it is by using soft prediction scores of the deep/teacher model as target labels to train the student/interpretable model. Student model would generally be a decision tree or forest based model or even rule-based model which could be later used for explaining predictions. As long as the approximation is sufficiently close, the statistical properties of the teacher model could be reflected in the student. Eventually, we obtain a model with comparable prediction performance, and the behavior of which is much easier to understand. Some work in this area includes, transforming a tree ensemble model into a single decision tree[108]. In addition, a deep neural net is utilized to train a single decision tree which mimics the function learnt by neural network so that the knowledge encoded in DNN is transferred to the decision tree [109] and also they employed active learning in their training pipeline to avoid overfitting of the decision tree. Overall, one advantage of mimic learning is that we still get comparable performance as that of complex model unlike local approximation based methods where we need to sacrifice more. *Local approximation* based methods [103, 104, 87, 98, 99] approximate the complex model with a simple model on the assumption that behaviors of complex model around the neighborhood of a given input is well approximated by the simple model [103]. We can use sparse linear model as our simple model and the weight vector of that linear model could be used to get feature contribution scores for the original complex model prediction. However, these explanations could be unfaithful in the cases where even the local behavior of complex model is extremely non-linear. Hence, for simple models, instead of using linear models, we use the ones that are able to capture the non-linear relationships. For example, a method can be designed using if-then rules [104] which are even able to explain both the current instance and some other relevant instances.

1.8 Challenges for Interpretable Detection

In general, there is a well known tradeoff between prediction accuracy and interpretability. The more interpretable models may result in reduced prediction accuracy compared to less interpretable ones [107, 3]. For example, a single decision tree would be more interpretable than a random forest. However, random forests models are known to achieve better prediction accuracy. Similarly, this can be generalized to several different models as shown in Fig. 1.7.

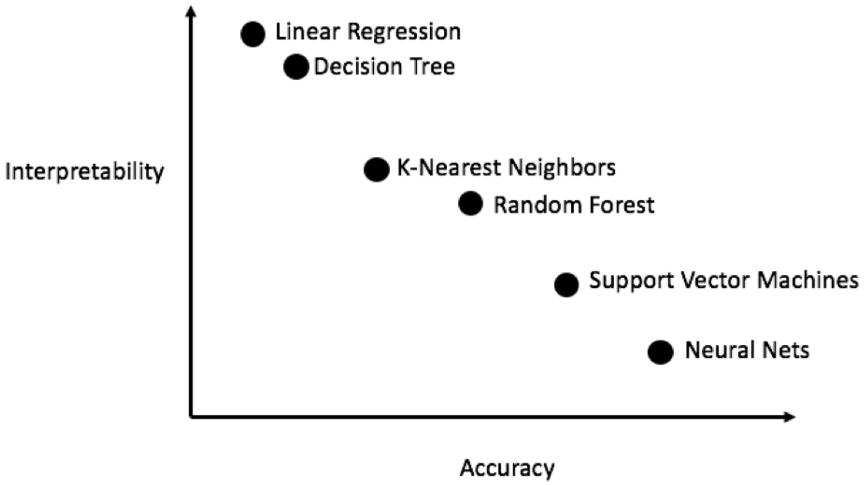


Figure 1.7: Comparison of predictive accuracy and interpretability of various machine learning methods. Neural nets typically have high accuracy and low interpretability.

1.9 Thesis Outline

In this thesis, we overcome the challenges mentioned in the above section by developing explainable fake news detection methods for vision and language that do not compromise much on accuracy. Specifically this thesis is organized in following chapters:

Chapter 2 focuses on developing explainable fake news detection method for vision based fake news which can predict and also reason its prediction in the form of heatmaps or class activation maps.

Chapter 3 focuses on correcting the above explanations to further refine the model generalization power. Also, this chapter introduces an active learning approach to make the refinement process feasible by significantly reducing the labeling efforts required for this refinement process.

Chapter 4 focuses on language based (text) fake news and develops three methods for explainable detection. Explanation for an instance prediction will include word/phrase importance, attribute importance and some supporting examples in the training data.

Chapter 5 introduces a web application for interactive explainable fake news detection for text based fake news. In the backend, this system uses algorithms developed in chapter 4 to predict and reason the user provided news input.

Chapter 6 offers discussion and future work.

2. EXPLAINING VISION BASED FAKE NEWS

Several manipulation methods are available on the web to create a fake image. However, manipulations that are noticeable to human eyes are less likely to create negative effects on the society. But manipulations with special characteristics when shared on social media could cause serious impact on our society. Two of those special characteristics include: *fine-grained nature* and *spatial locality*. First says that the difference between true and fake images are so subtle, even human eyes are hard to distinguish them. Second says that the forgery occupies only a certain ratio of the whole image input. For instance, DeepFake videos [11] use GAN-based technology to replace one's face with anther's. This manipulation changes human faces, while leaving the background part unchanged. Considering these two characteristics, a desirable detection model should be able to concentrate on the forgery region to learn effective representations. As such, the detection model needs to possess local interpretability, which indicates which region is attended by the model to make decisions [3]. The benefit is that we can later control the local interpretation explicitly by imposing extra supervision on instance interpretation in the learning process, inorder to enforce the model to focus on the forgery region to learn representations.

2.1 Detecting Fake Images

Our objective here is to train a network, which could distinguish fake images from true ones. A key characteristic of forgery detection lies in its fine-grained nature. Thus effective representation is needed for both true and fake images in order to ensure high detection accuracy. As such, we use an autoencoder to learn more distinguishable representations which could separate true and fake images in the latent space.

2.1.1 Notations

Here we introduce the basic notations used in this section. Given a *source dataset* $\mathcal{D}$ containing both true images $\mathbf{X}_T$ and fake images $\mathbf{X}_F$ generated by a forgery method. $\mathcal{D}$ is split into training set $\mathcal{D}_{\text{trn}} = \{(x_i, l_i)\}_{i=1}^{N}$, validation set $\mathcal{D}_{\text{val}} = \{(x_i, l_i)\}_{i=1}^{N_{\text{val}}}$ and test set $\mathcal{D}_{\text{tst}} = \{(x_i, l_i)\}_{i=1}^{N_{\text{tst}}}$, where

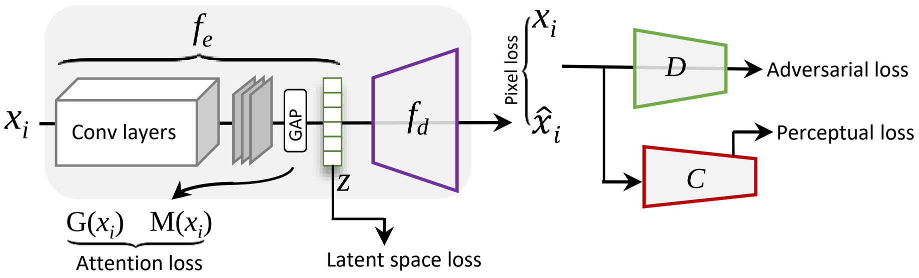


Figure 2.1: Overview of our fake image detection framework AE.

$l_i \in [0, 1]$ denotes fake and true class label respectively. A detection model $f(x)$ is learned from the training set $\mathcal{D}_{\text{trn}}$ and evaluated on test set $\mathcal{D}_{\text{tst}}$. Validation set $\mathcal{D}_{\text{val}}$ is used for early stopping or model selection use case.

2.1.2 Methodology

The autoencoder (AE) is denoted using f, which consists a sub-network encoder $f_e(\cdot)$ and decoder $f_d(\cdot)$. This encoder maps the input image $x \in R^{w\times h\times 3}$ to the low-dimensional latent vector space encoding $z \in R^{d_z}$, where d_z is the dimension of latent vector z. Then the decoder remaps latent vector z back to the input space $\hat{x} \in R^{w\times h\times 3}$. Both operations can be represented mathematically as below.

$$z = f_e(x, \theta_e), \quad \hat{x} = f_d(z, \theta_d), \tag{2.1}$$

where θ_e and θ_d are parameters for the encoder and decoder respectively. To enforce our model to learn more meaningful and intrinsic features, we introduce the latent space loss as well as reconstruction loss.

$$\mathcal{L}_1(\theta_e, \theta_d, x, l) = \mathcal{L}_{\text{rec}} + \mathcal{L}_{\text{latent}}. \tag{2.2}$$

The autoencoder (AE) is visualized in Fig. 2.1. Typically, encoder and decoder would be a sequence of conv/deconv layers with pooling, RELU operations. The specific network architecture we used is listed under experiments section (2.1.3). The motivation behind using Global Average Pooling (GAP) layer after last convolution layer in the encoder is illustrated under explaining the detections section (2.2). Above two loss functions are elaborated below.

Latent Space Loss: We make use of the latent space representation to distinguish the forgery images from the true ones. The latent space vector is first split into two parts: $T = \{1, ..., \frac{d_z}{2}\}$, and $F = \{\frac{d_z+2}{2}, ..., d_z\}$. The total activation of x_i for the true and fake category respectively is denoted as:

$$a_{i,T} = \frac{2}{d_z}||z_{i,c}||_1,\ c \in T;\ a_{i,F} = \frac{2}{d_z}||z_{i,c}||_1,\ c \in F. \tag{2.3}$$

The final latent space loss is defined as follows:

$$\mathcal{L}_{\text{latent}} = \sum_i |a_{i,T} - l_i| + |a_{i,F} - (1 - l_i)|, \tag{2.4}$$

where l_i is the ground truth of input image x_i. The key idea of this loss is to enforce the activation of the true part: $\{z_{i,c}\},\ c \in T$ to be maximally activated if the input x_i is a true image, and similarly to increase the fake part $\{z_{i,c}\},\ c \in F$ activation values for fake image inputs. At testing stage, the forgery detection is based on the activation value of the latent space partitions. The input image x_i is considered to be true if $a_{i,T} > a_{i,F}$, and vice versa.

Reconstruction Loss: To force the fake and true images more distinguishable in the latent space, it is essential to learn effective representations. Specifically, we use reconstruction loss which contains three parts: *pixel-wise loss*, *perceptual loss*, and *adversarial loss*, to learn intrinsic representation for all training samples. The overall reconstruction loss $\mathcal{L}_{\text{rec}}$ is defined as follows:

$$\sum_i \beta_1 \underbrace{||x_i - \hat{x}_i||_2^2}_{\text{Pixel Loss}} + \beta_2 \underbrace{||C(x_i) - C(\hat{x}_i)||_2^2}_{\text{Perceptual Loss}} + \beta_3 \underbrace{[-\log(D(\hat{x}_i))]}_{\text{Adversarial Loss}}. \tag{2.5}$$

The pixel-wise loss is measured using mean absolute error (MAE) between original input image pixels and reconstructed image pixels. For perceptual loss, a pretrained comparator $C(\cdot)$ (e.g., VGGNet [110]) is used to map input image to feature space: $R^{w \times h \times 3} \rightarrow R^{w_1 \times h_1 \times d_1}$. Then MAE difference at the feature space is calculated, which represents high-level semantic difference of x_i and $\hat{x}_i$. In terms of adversarial loss, a discriminator $D(\cdot)$ is introduced aiming to discriminate the generated images $\hat{x}_i$ from real ones x_i. This subnetwork $D(\cdot)$ is the standard discriminator network introduced in DCGAN [111], and is trained concurrently with our autoencoder. The autoencoder is trained to trick the discriminator network into classifying the generated images as real. The discriminator D is trained using the following objective:

$$\mathcal{L}_D = -[E_{x \sim P_X}[\log D(x)] + E_{x \sim P_X}[\log(1 - D(\hat{x}))]]. \tag{2.6}$$

Parameter β_1, β_2, β_3 are employed to adjust the impact of invidual losses. The three losses serve the purpose of ensuring reconstructed image to: 1) be sound in pixel space, 2) be reliable in the high-level feature space, and 3) look realistic respectively. The implicit effect is to force the vector z to learn intrinsic representation which could make it better separate fake and true images. Besides, using three losses instead of using only pixel-wise loss could help stabilize the training in less number of epochs [112].

We conduct several experiments using the overall loss $\mathcal{L}_1$ defined in Eq.(2.2) which showed it's effectiveness across different kinds of manipulations.

2.1.3 Experiments

We conduct experiments to answer the following research questions. (1) Does the proposed AE achieve better accuracy compared to other state-of-the-art methods? (2) How do different components and hyperparameters affect the performance of AE?

In this section, we introduce the overall experimental setups including datasets, baselines, networks architecture and implementation details.

Datasets: We have considered two types of manipulations that modify true images. For each manipulation, we have considered two different methods to generate fake images based on their true versions. Manipulation types and datasets creation process is illustrated below. We will mainly focus on creating fake images by doing modifications to face, because of their likelihood to have high impact on society.

- **Computer Graphics based manipulation.** We have used a public database FaceForensics++, proposed in [113], with 1000 real videos and 2000 manipulated videos. Among 2000 fake videos, 1000 are created with the method Face2Face [114] and 1000 with the method

	Face		**Attribute**		**Inpainting**	
	Face2face	FaceSwap	Glow	StarGAN	G&L	ContextAtten
Train	288000	288000	41590	41590	28000	28000
Val	2800	2800	11952	11952	6000	6000
Test	2800	2800	5982	5982	6000	6000

Table 2.1: Dataset statistics for computer graphics based manipulation (includes face modification), deep learning based manipulation (includes attribute modification, inpainting-based modification).

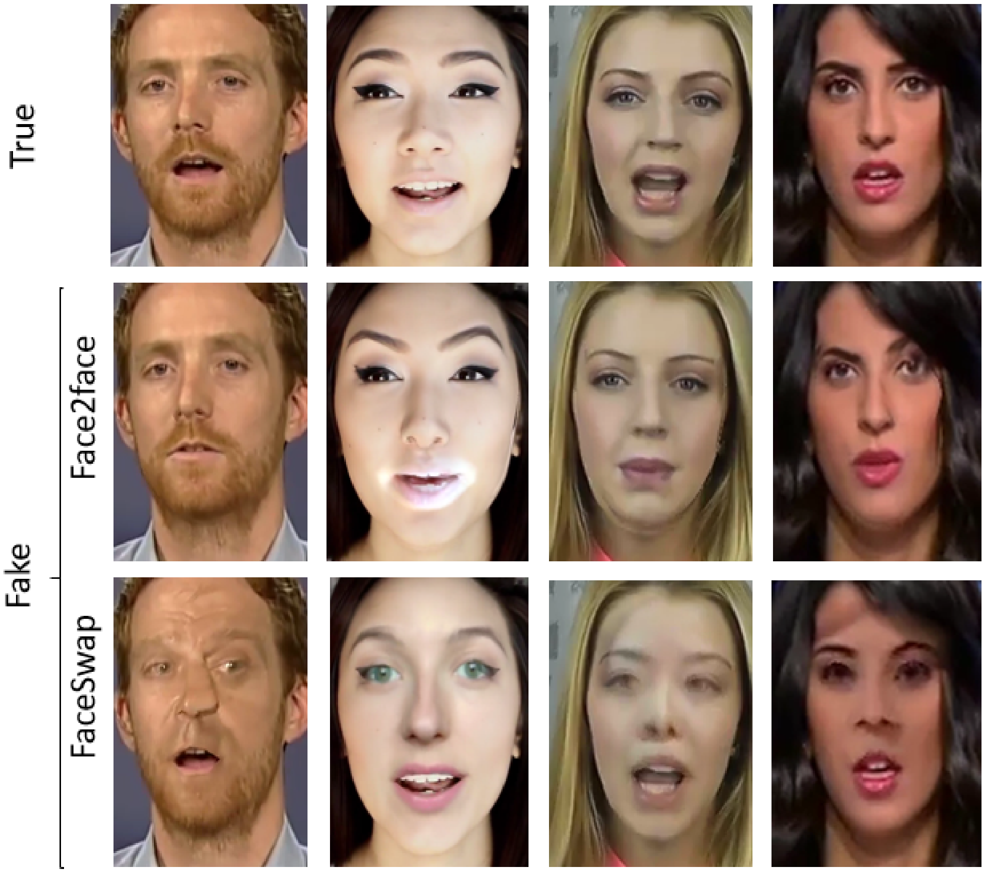


Figure 2.2: Examples of computer graphics based manipulations to facial region. Top row contain original images and later rows contain fake images created using Face2face and FaceSwap methods respectively.

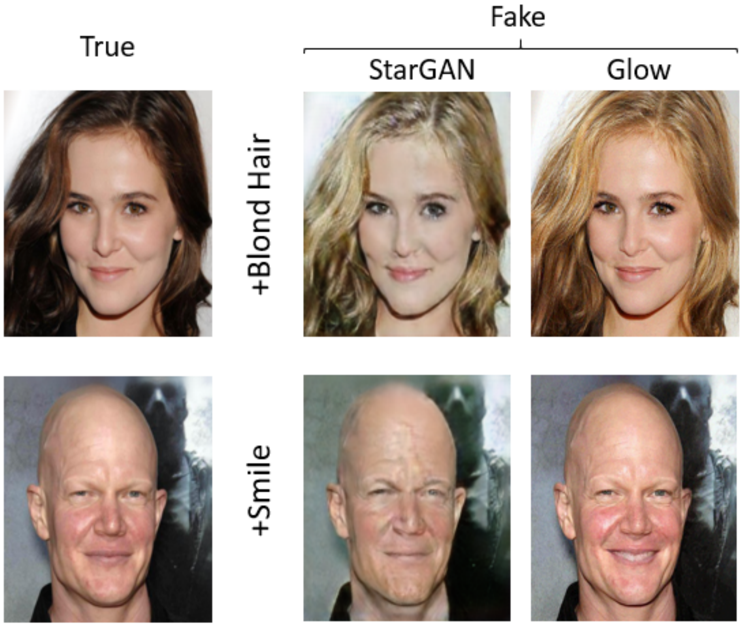


Figure 2.3: Examples of deep learning based manipulations to modify facial attributes. Each row contain a true image and two fake images created using StarGAN and Glow methods respectively. Attribute name is included inbetween true and fake images.

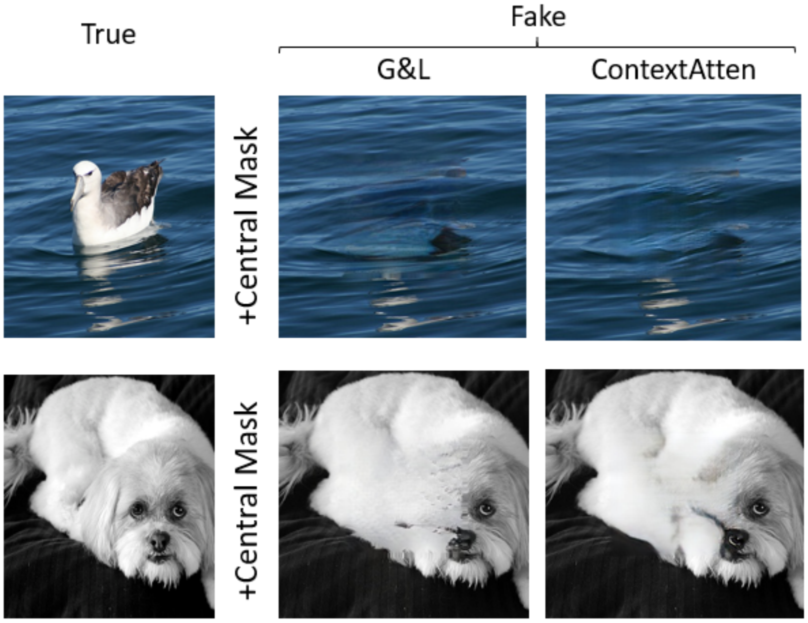


Figure 2.4: Examples of deep learning based manipulations applied to the central region of the image using inpainting based methods. Each row contain a true image and two fake images created using G&L and ContextAtten methods respectively.

FaceSwap [115]. Both the methods are graphics based and yield different datasets. In general, Face2Face generates fake images that are particularly harder to detect by human observers[113] since Face2Face does not introduce a strong semantical change, and thus, introducing only subtle visual artifacts in comparison to the face replacement methods like FaceSwap. Each dataset was split into 704 videos for training, 150 for validation, and 150 for testing. All splits are balanced, where the ratio of true and fake images are 1:1. All videos have been compressed using H.264 with quantization parameter set to 23. Images were extracted from videos using Cozzolino et al.'s settings [55]: 200 frames of each training video were used for training, and 10 frames of each validation and testing video were used for validation and testing, respectively. There is no detailed description of the rules for frame selection, so we selected the first (200 for train or 10 for test/val) frames of each video and cropped the facial areas based on the segmentation masks provided in the database for all manipulated videos. This public database also comes with ground truth masks for all fake images that indicate whether a pixel has been modified or not, which can be used to train forgery localization methods. Corresponding dataset statistics are given in Tab. 2.1 and images are shown in Fig. 2.2.

- **Deep Learning based manipulation.** With advancements of deep learning techniques, it is now possible to generate super-realistic fake images. Here we consider two categories of datasets [55], fake images created using GAN-based attribute modification and fake images created from inpainting-based modification. Under the first category, real images from CelebA dataset [116] are modified with two methods: StarGAN [117] and Glow [118]. The modified attributes include changing hair color, changing smile, etc. Some examples are shown in Fig. 2.3. Under the second category, original images are modified using two inpainting methods, G&L [119] and ContextAtten [120]. The inpainting is performed to central 128×128 pixels of the original images as shown in Fig. 2.4. For both the categories, all images are 256×256 pixels and all splits are balanced, where the ratio of true and fake images are 1:1.

Preprocessing: For all datasets, we have applied normalization with mean (0.485, 0.456, 0.406), standard deviation (0.229, 0.224, 0.225), since these values have been widely used in the ImageNet Large Scale Visual Recognition Challenge [121]. We haven't applied any data augmentation in our experiments. If needed, images has been resized to meet network input dimension requirements using a bilinear interpolation.

Baselines: We evaluate AE by comparing it with six baselines. All models are trained on the same data and evaluated on the same data.

- **SuppressNet** [122]: A generic manipulation detector that uses a constrained convolutional layer followed by two convolutional, two max-pooling and three fully-connected layers. Constrained convolution layer is designed to suppress the high-level contents of the image.

- **ResidualNet** [123]: Residual-based descriptors are used for forgery detection. This model recasts the hand-crafted Steganalysis features used in the forensic community to a CNN-based network. Basically, these features are extracted as co-occurrences on 4 pixels patterns along horizontal and vertical direction on the residual image, which is obtained after high-pass filtering of the original input image.

- **StatsNet** [124]: To optimize the feature extraction scheme, this method integrates the computation of statistical feature extraction within a CNN framework. CNN framework consists of a global pooling layer that computes four statistics (mean, variance, maximum, minimum). We consider the Stats-2L network since this model has the best performance.

- **MesoInception** [50]: This is a CNN-based network specifically designed to detect face manipulations in videos. It uses two inception modules, two convolution layers with max-pooling, followed by two fully-connected layers at the end. Mean square error instead of cross-entropy is used as loss function.

- **XceptionNet** [125]: A CNN based network, where depth-wise separable convolution layers with residual connections is used for forgery detection. We use a pretrained network on

Encoder layer	**Output shape**	**Decoder layer**	**Output shape**
Conv2d	[64, 128,128]	ConvTranspose2d	[256, 4,4]
Relu	[64, 128,128]	BatchNorm2d & Relu	[256, 4,4]
Conv2d	[128,64,64]	ConvTranspose2d	[128, 8,8]
BatchNorm2d	[128,64,64]	BatchNorm2d & Relu	[128, 8,8]
Relu	[128,64,64]	ConvTranspose2d	[64, 16,16]
Conv2d	[256,32,32]	BatchNorm2d & Relu	[64, 16,16]
BatchNorm2d	[256,32,32]	ConvTranspose2d	[32, 32,32]
Relu	[256,32,32]	BatchNorm2d & Relu	[32, 32,32]
Conv2d	[512,16,16]	ConvTranspose2d	[16, 64,64]
BatchNorm2d	[512,16,16]	BatchNorm2d & Relu	[16, 64,64]
Relu	[512,16,16]	ConvTranspose2d	[8, 128,128]
Conv2d	[512,16,16]	BatchNorm2d & Relu	[8, 128,128]
Relu	[512,16,16]	ConvTranspose2d	[3, 256,256]
AvgPool2d	[512,1,1]	Tanh	[3, 256,256]
Linear	[128]		

Table 2.2: Network architecture and output shapes.

ImageNet by replacing last fully connected layer with two outputs in order to match our use-case. We use ImageNet weights to initialize all other layers. To set up the newly inserted fully connected layer, we fix all weights up to this new layer and pre-train the network for 3 epochs. Finally, we train the network for additional 20 epochs and choose the one with with best accuracy on validation set.

- **ForensicTransfer** [55]: This is an encoder-decoder based architecture with 5 convolution layers in each sub-network. Decoder additionally uses a 2 × 2 nearest-neighbor up-sampling before each convolution (except the last one) to recover the original size. The latent space (encoder output) has 128 feature maps among which 64 are associated with the real class and 64 with the fake class. For a fair comparison, we use their version that is not fine-tuned on target dataset.

Network Architecture: For encoder and decoder, we use a structure similar to U-net [126].

Details about the layers and corresponding output shapes are given in Tab. 2.2. The AvgPool2d corresponds to global average pooling layer, which transform the [512,16,16] activation layer into 512 dimension vector. After that, we use a Linear layer to turn it into the 128-dimension latent space vector z (see Fig. 2.1). For comparator $C(\cdot)$, we use the 16-layer version VGGNet [110], and the activation after 10-th convolutional layer with output shape [512,28,28] is used to calculate the perceptual loss. For discriminator D, we use the standard discriminator network introduced in DCGAN [111].

Training Details: For fair comparison, all models are trained on the same data and tested on the same test set. We have used early stopping strategy using validation loss as stopping criteria with patience set to 10 and for a maximum of 40 epochs. Model specific training details are illustrated below.

- **AE.** For training, we use the Adam optimizer [127] with a learning rate of 0.001, batchsize of 64 and default values for the moments $\beta 1$ = 0.9, $\beta 2$ = 0.999 and epsilon set to 10^{-8}. Tuned all three hyperparameters used in the equations in the methodology. For the first two hyperparameters (β_1, β_2) in Eq.(2.5), we have tuned values between 0 and 1 with 0.1 as interval and for the third (β_3), we have tried {0.0001, 0.001, 0.01, 0.05, 0.1, 0.5, 1.0}. During evaluation, we only use encoder part to make the decision.

- **Baselines.** All baselines are trained using the same loss optimizations mentioned in their papers. Regarding hyperparameters, for SupressNet [122] and ResidualNet [123], we use a learning rate $= 10^{-5}$ with batch size of 64 and 16 respectively. For StatsNet [124] and MesoInception [50], we use batch size of 64 with learning rates of 10^{-4}, 10^{-3} respectively. XceptionNet [125] is trained with batch size of 32 and learning rate of 0.0002. For ForensicTransfer [55], we use a learning rate of 0.001 and a batch size of 64.

Accuracy Evaluation: For all datasets, detection accuracy on corresponding test sets is used as metric to compare the models. For each dataset, training is done on train set and evaluated

	Face		Attribute		Inpainting	
Models	Face2face	FaceSwap	StarGAN	Glow	G&L	ContextAtten
SuppressNet	93.86	93.04	99.98	98.94	99.08	99.06
ResidualNet	86.67	87.74	99.98	98.87	98.96	**99.24**
StatsNet	92.94	91.69	99.98	99.01	96.17	94.74
MesoInception	94.38	92.52	**100.0**	**99.04**	86.90	95.37
XceptionNet	**98.02**	**98.67**	**100.0**	98.98	**99.86**	98.12
ForensicTransfer	93.91	94.16	**100.0**	**99.04**	99.65	99.01
AE	96.92	97.08	**100.00**	**99.04**	99.74	99.12

Table 2.3: Detection accuracy of several models for six datasets. Values are reported based on their performance on corresponding test set. All models are trained on dataset specific train set and evaluated on dataset specific test set.

on corresponding test set. For example, results on Face2face dataset indicate performance on Face2face test set after training on Face2face train set. Tab. 2.3 shows the comparison between our method AE and several baselines across six datasets. For AE, following hyperparameters gave the best results: $\alpha_1 = 1.0$, $\alpha_2 = 1.0$, $\beta_1 = 1.0$, $\beta_2 = 1.0$, $\beta_3 = 0.01$. In general, there are two interesting observations.

1. Models with latent space loss and lesser number of parameters (AE, ForensicTransfer) perform almost as good as pretrained/very deep layered models like XceptionNet or MesoInception. Additional adversarial, perceptual losses in our model (AE) have helped outperform ForensicTransfer model. We will further stretch about the impact of different losses for AE under ablation and hyperparameter analysis section.

2. On all datasets, most of the models are giving close to 100% accuracy which can trigger a doubt on generalization ability. It is important to think of generalization as it's difficult to come up with individual model for each forgery method like Face2Face, FaceSwap etc.

Ablation and Hyperparameters Analysis: We utilize our model (AE) trained on face manipulation task to conduct ablation and hyperparameter analysis which could provide more insights about the contribution of different components in AE.

	AE_rec	AE_latent	AE_latent_pixel	AE
Face2face	50.39	95.82	96.57	96.92
FaceSwap	58.12	93.24	95.10	97.08

Table 2.4: Ablation analysis of AE for face manipulation task.

	β_1	β_2	β_3	Face2face	FaceSwap
Alter pixel	**1.0**	1.0	0.01	96.92	97.08
	0.5	1.0	0.01	96.01	95.07
	0.1	1.0	0.01	95.55	94.54
Alter perceptual	1.0	**1.0**	0.01	96.92	97.08
	1.0	**0.5**	0.01	96.74	95.92
	1.0	**0.1**	0.01	95.84	93.67
Alter adversarial	1.0	1.0	**0.1**	54.16	65.09
	1.0	1.0	**0.05**	58.28	76.12
	1.0	1.0	**0.01**	96.92	97.08

Table 2.5: Hyperparameter analysis for $\beta_1, \beta_2, \beta_3$

- **Ablation analysis.** We compare AE with its ablations to identify the contributions of different components. Four ablations include: AE_rec, trained only with reconstruction loss of Eq.(2.5); AE_latent, using only latent space loss in Eq.(2.3); AE_latent_pixel, using both latent space loss and pixel loss in Eq.(2.5); AE_latent_rec, using latent space loss and whole reconstruction loss in Eq.(2.2). Note that no attention loss is used in the ablations. The comparison results are given in Tab. 2.4. There are several key findings. Firstly, latent space loss is the most important part, without which even source test set accuracy could drop to 50.39% on Face2face and 58.12% on FaceSwap. Secondly, all of pixel-wise, perceptual, and adversarial losses could contribute to additional performance boost.

- **Hyperparameters analysis.** We evaluate the effect of different hyperparameters towards model performance by altering the values of $\beta_1, \beta_2, \beta_3$ in Eq.(2.5). Corresponding results are reported in Tab. 2.5. The results indicate that increase of weights for pixel loss and perceptual loss could enhance model performance. In contrast, a small weight for adversarial loss is beneficial for accuracy improvement. Also, reconstuction loss is more important to FaceSwap than Face2Face which indeed make sense as Face2face way of creating fake images do not introduce a strong semantic change, introducing only subtle visual artifacts in contrast to FaceSwap [113].

2.2 Explaining the Detections

To build trust on deep learning methods, reasoning their predictions could be very useful. Although, our autoencoder AE developed in last section gives more than 96% accuracy on all datasets, due to data-driven training paradigm, there it's not guaranteed that our model focuses on the forgery region to make predictions, instead might have learnt spurious correlations by capturing biased artifacts in the dataset. To validate this claim, first we need to generate some explanation for every prediction.

There are couple of differnt ways to generate heatmaps to explain a CNN prediction as described in chapter 1. We chose CAM explanation method because it is end-to-end differentiable, amenable for training with backpropagation and updating CNN parameters. Detailed illustration of generating explanations in our case is given below.

2.2.1 Methodology for Local Interpretation

This is just a post-hoc way of generating heatmaps so none of our model AE parameters are modified. The goal of local interpretation is to identify the contributions of each pixel in the input image towards a specific model prediction [86]. The interpretation is illustrated in the format of heatmap (or attention map). Inspired by the CNN local interpretation method Class Activation Map (CAM) [88], we use global average pooling (GAP) layer as ingredient in the encoder, as illustrated in Fig. 2.1. This enables the encoder to output attention map for each input. Let l-layer denotes the last convolutional layer of the encoder, and $f_{l,k}(x_i)$ represents the activation matrix at k-channel of l-layer for input image x_i. Let also w_k^c corresponds to the weight of k-channel towards the unit c of latent vector z. The CAM attention map for unit c is defined as follows:

$$M_c(x_i) = \sum_{k=1}^{d_l} w_k^c \cdot f_{l,k}(x_i). \tag{2.7}$$

Later we upsample $M_c(x_i)$ to the same dimension as the input image x_i using bilinear interpolation. Each entry within $M_c(x_i)$ directly indicates the importance of the value at that spatial grid of image x_i leading to the activation z_c. The final attention map $\hat{M}(x_i)$ for an input image x_i is denoted as:

$$\hat{M}(x_i) = \sum_{c=1}^{d_F} |z_{i,c}| \cdot M_c(x_i) = \sum_{c=1}^{d_F} \sum_{k=1}^{d_l} |z_{i,c}| \cdot w_k^c \cdot f_{l,k}(x_i), \tag{2.8}$$

where $z_{i,c}$ denotes the c-th unit of the latent vector z for x_i.

2.2.2 Visualizations of Interpretation

In this section, we provide heatmap visualizations of AE predictions on Face manipulation dataset created using Face2Face.

We have generated explanations for trained AE model using Eq.(2.7). This process will not change any of our model parameters. These explanations can help in developing trust on our AI system. For each prediction of AE model, corresponding explanations as shown in Fig. 2.5. On

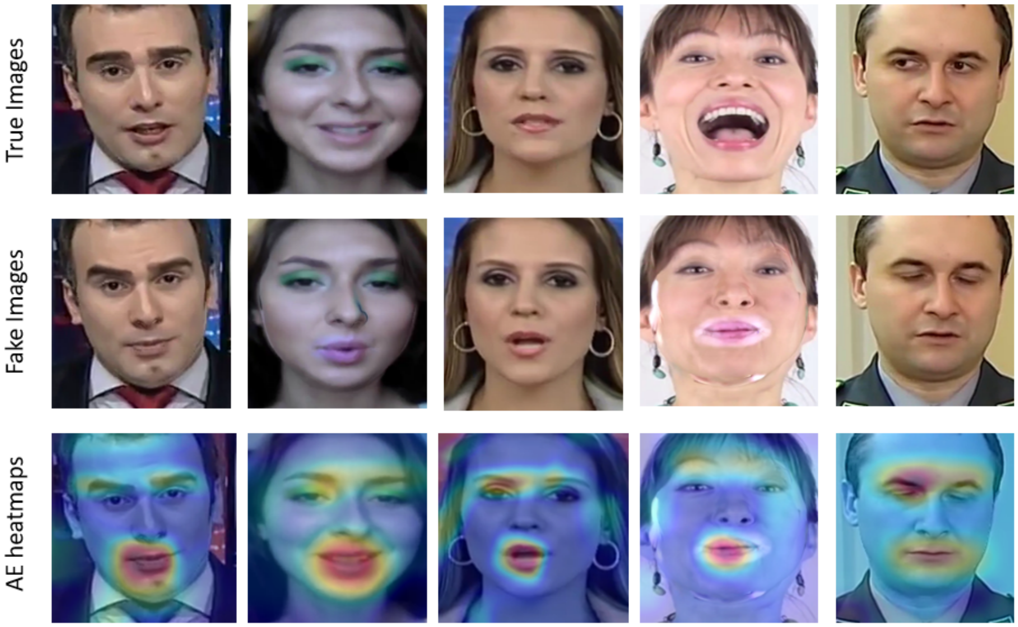


Figure 2.5: Examples of heatmaps generated by AE for face manipulation data. Top row contain original images, second row contain fake images created using Face2face and last are the corresponding explanations of AE model.

the other hand, we can notice that model predictions are not fully focused on foregery part, indeed capturing background information, important artifacts in the dataset. For Face2Face generated fake images, most of the explanations are focused on artifacts like eye brows, mouth as shown in Fig. 2.5. Although these are indeed modified artifacts in most of the images, but not necessarily true for all the images. For example, in the above figure, regions like eyes (in last column image), smile (in 4th column image) remain unimportant for making the decision. Also, originally unmodified regions like eyebrows (in column 3, last column images) are being important for the model decision. Although model correctly predicted all these images as fake, explanations do not make sense to humans for images in last three columns.

We have further stretched on this limitation in chapter 3 and shown the benefits of correcting these explanations when building global models.

3. USING EXPLANATIONS TO ENHANCE GENERALIZATION

Due to limitations in getting annotated data for every task in NLP/Vision, there has been lot of interest in building a global/multi-task models in both language and vision communities. If that's the case, then generalization ability would be a key factor during model selection stage. Most of the current deep learning methods employ pure data-driven training paradigms. Therefore, it's very likely for them to capture biases or certain spurious correlations which happen to be predictive in the current dataset. For example, our fake image detection model chapter 2 have shown more than 96% acccuracies on all our datasets. But most of their explanations in chapter 2 do not focus on the correct forgery part rather captured important artifacts that are sufficient to make a correct decision. However, accuracies can drop to 50% if that trained model is tested on related datasets [55]. Recent work of [55] shown that using a model trained on Face2Face generated images to evaluate on FaceSwap generated images have reduced the accuracy to almost 50% which is similar to random guessing. This kind of overfitting is a serious issue if we plan to use such models on real world data.

In this chapter, we restrict our focus to vision and propose a method to improve the generalization ability of our detection model AE developed in chapter 2. Later, we shown that our newly proposed method makes predictions relying on correct evidence and also achieves state-of-the-art generalization performance on all datasets with improved interpretability.

3.1 Locality-aware AutoEncoder (LAE)

The key idea of LAE is that the model should focus on correct regions and exploit reasonable evidences rather than capture biases within dataset to make predictions. Due to the pure data-driven training paradigm, the autoencoder AE developed in section chapter 2 is not guaranteed to focus on the forgery region to make predictions. Instead the AE may capture certain spurious correlations which happen to be predictive in the current dataset. This would lead to decreased generalization performance on unseen data generated by alternativeforgery methods. In LAE (as illustrated in dcgan), we explicitly enforce the model to rely on the forgery region to make detection predictions, by augmenting the model with local interpretability developed in chapter 2 and regularizing the interpretation with extra supervision. Besides, we design an active leaning framework to select the challenging candidates for regularizing LAE.

Augmenting Local Interpretability: This approach is same as the way we generated heatmaps in sec 2.2.1. The goal here is to identify the contributions of each pixel in the input image towards a specific model prediction [86]. The interpretation is illustrated in the format of heatmap (or attention map). Inspired by the CNN local interpretation method Class Activation Map (CAM) [88], we use global average pooling (GAP) layer as ingredient in the encoder, as illustrated in Fig. 2.1. This enables the encoder to output attention map for each input. Let l-layer denotes the last convolutional layer of the encoder, and $f_{l,k}(x_i)$ represents the activation matrix at k-channel of l-layer for input image x_i. Let also w_k^c corresponds to the weight of k-channel towards the unit c of latent vector z. The CAM attention map for unit c is defined as follows:

$$M_c(x_i) = \sum_{k=1}^{d_l} w_k^c \cdot f_{l,k}(x_i). \tag{3.1}$$

Later we upsample $M_c(x_i)$ to the same dimension as the input image x_i using bilinear interpolation. Each entry within $M_c(x_i)$ directly indicates the importance of the value at that spatial grid of image x_i leading to the activation z_c. The final attention map $\hat{M}(x_i)$ for an input image x_i is denoted as:

$$\hat{M}(x_i) = \sum_{c=1}^{d_F} |z_{i,c}| \cdot M_c(x_i) = \sum_{c=1}^{d_F} \sum_{k=1}^{d_l} |z_{i,c}| \cdot w_k^c \cdot f_{l,k}(x_i), \tag{3.2}$$

where $z_{i,c}$ denotes the c-th unit of the latent vector z for x_i.

Regularizing Local Interpretation: To enforce the network to focus on the correct forgery region to make detection, a straightforward way is to use instance-level forgery ground truth to regularize the local interpretation. Specifically the regularization is achieved by minimizing the distance between individual interpretation map $\hat{M}(x_i)$ and the extra supervision for all the N_F

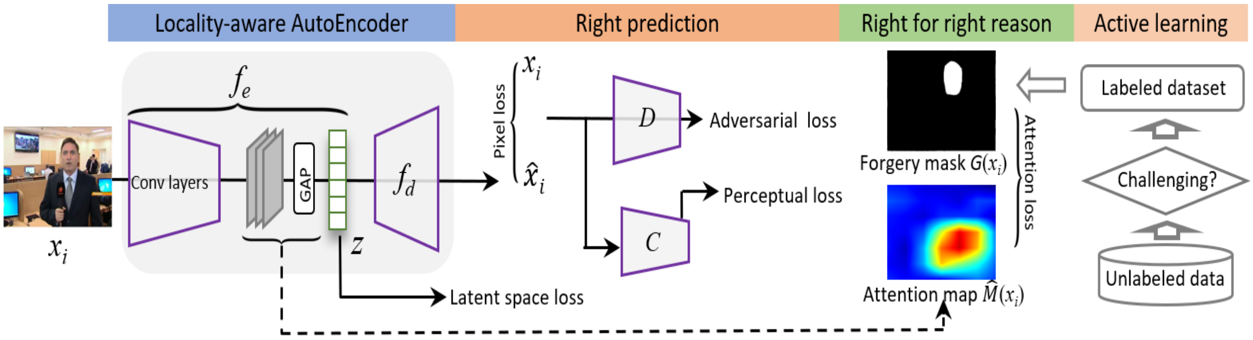


Figure 3.1: Schematic of LAE training for generalizable forgery detection. Latent space and reconstruction losses to learn effective representation; extra supervision to regularize heatmap to boost generalization accuracy; active learning to reduce forgery masks annotation efforts. Note that the main difference between AE and LAE is the attention loss in Eq. 3.3

forgery images. The attention loss is defined as follows:

$$\mathcal{L}_{\text{attention}}(\theta_e, x, G) = \sum_{i=1}^{N_F}[\hat{M}(x_i) - G(x_i)]^2, \tag{3.3}$$

where $G(x_i)$ denotes extra supervision, which is annotated ground truth for forgery. This ground truth is given in the format of pixel-wise binary segmentation mask (see Fig. 3.1 for an illustrative example). The attention loss is end-to-end trainable and can be utilized to update the model parameters. Ultimately the trained model could focus on the manipulated regions to make decisions.

However, getting annotated forgery masks is time consuming especially for big datasets. So below we present an active learning framework to reduce the annotation efforts and later, shown that regularizing AE with just less than 1% annotations can boost generalization accuracy if those 1% annotations are the candidates filtered by our active learning framework.

3.2 Active Learning to Regularize Local Interpretation

However, generating pixel-wise segmentation masks is extremely time consuming, especially if we plan to label all N_F forgery images within a dataset. We are interested in employing only a small ratio of data with extra supervision. In this section, we propose an active learning framework to select challenging candidates for annotation. At each iteration, we select a a small ratio of data has ground truth for the forgery region through the following two steps iteratively. We will describe below how the active learning works in three steps. **Channels concept ranking:** Due to the hierarchical structure of encoder, the last convolutional layer has larger possibility to capture high-level semantic concepts. In our case, we have 512 channels at this layer. A desirable detector could possess some channels which are responsive to specific and semantically meaningful natural part(e.g., face, mouth, or eye), while other channels may capture concepts related to forgery, (e.g., warping artifacts, or contextual inconsistency). Nevertheless, in practice the detector may rely on some spurious patterns which only exist in the training set to make forgery predictions. Those samples leading to this concept are considered as the most challenging case, since they cause the model to overfit to dataset specific bias and artifacts.

We intend to select out a subset of channels in the last convolutional layer deemed as most influential to the forgery classification decision. The contribution of a channel towards a decision is defined as the channel's average activation scores for an image. Specifically, the contribution of channel k towards image x_i is denoted as: $\{u_{i,k}\}_{k=1}^{d_c}$, where d_c is the number of channels. We learn a linear model based on the d_c concepts to predict the possibility of image x_i to be fake: $p(u_i) = \frac{\exp(w \cdot u_i)}{1+\exp(w \cdot u_i)}$. The loss function is defined as:

$$\mathcal{L}_w = \sum_{i=1}[l_i \cdot \log(p(u_i)) + (1 - l_i) \cdot \log(1 - p(u_i))]. \tag{3.4}$$

After this training, we select 10 highest components of the optimized linear weight vector w and the corresponding channels are considered as more relevant to the forgery decision.

Active candidate selection: After locating the most possible channels corresponding to the forgery prediction, we feed all the N_F fake images to the LAE model. Those who have highest activation value for these top 10 channels are deemed as the challenging case. The key idea for this choice is that these highest activation images are mostly likely to contain easy patterns which can be captured by the model to separate true and fake images, and which are hard to be generalized

Algorithm 1: Locality-aware AutoEncoder (LAE).
Note: Steps 1 to 6 are same as our AE model

```
   Input: Training data $D = \{(x_i, l_i)\}_{i=1}^{N}$.
1  Set hyperparameters $\alpha_1, \alpha_2, \beta_1, \beta_2, \beta_3, \lambda_1, \lambda_2$, learning rate $\eta$, iteration number
     $max\_iter1, max\_iter2$, epoch index $t = 0$;
2  Initialize autoencoder parameters $\theta_e, \theta_d$;
3  while $t \leq max\_iter1$ do
4      $\mathcal{L}_1(\theta_e, \theta_d, x, l) = \alpha_1 \mathcal{L}_{\text{rec}} + \alpha_2 \mathcal{L}_{\text{latent}}$;
5      $\theta_{e,t+1}, \theta_{d,t+1} = Adam(\mathcal{L}_1(\theta_e, \theta_d, x, l), \eta)$;
6      $t = t + 1$;
7  Reduce the learning rate: $\eta \leftarrow \frac{\eta}{10}, t \leftarrow 0$;
8  while $t \leq max\_iter2$ do
9      $\mathcal{L}_w = \sum_{i=1}[l_i \cdot \log(p(u_i)) + (1 - l_i) \cdot \log(1 - p(u_i))]$;
10     Select out $N_{\text{active}}$ images as active candidates;
11     Request labeling pixel-wise masks $\{G(x_i)\}_{i=1}^{N_{\text{active}}}$;
12     $\mathcal{L}_{\text{attention}}(\theta_e, x, G) = \sum_{i=1}^{N_{\text{active}}}[\hat{M}(x_i) - G(x_i)]^2$;
13     $\mathcal{L}_2(\theta_e, x, l, G) = \lambda_1 \mathcal{L}_{\text{latent}} + \lambda_2 \mathcal{L}_{\text{attention}}$;
14     $\theta_{e,t+1} = Adam(\mathcal{L}_2(\theta_e, x, l, G), \eta)$;
15     $t = t + 1$; $\eta \leftarrow \frac{\eta}{10}$ if $t$ % 3 = 0;
   Output: LAE makes right predictions based on right reasons.
```

beyond training and hold-out test set. Thus we would like to request their pixel-wise forgery masks and followed by regularizing them. Based on this criteria, we select out N_{active} images as active candidates. The candidates number N_{active} is less than 1% of total images and is empirically shown significant improvement on generalization accuracy. Comparing to the number of total training samples which is larger than 10k, we have dramatically reduced the labelling efforts.

Local interpretation loss: Equipped with the active image candidates, we request labeling those images for pixel-wise forgery masks $\{G(x_i)\}_{i=1}^{N_{\text{active}}}$. The attention loss is calculated using the distance between interpretation map and annotated forgery mask for all N_{active} candidate images, which is further combined with latent space loss to update model parameters.

$$\begin{gathered}\mathcal{L}_{\text{attention}}(\theta_e, x, G) = \sum_{i=1}^{N_{\text{active}}} [\hat{M}(x_i) - G(x_i)]^2, \\ \mathcal{L}_2(\theta_e, x, l, G) = \lambda_1 \mathcal{L}_{\text{latent}} + \lambda_2 \mathcal{L}_{\text{attention}}\end{gathered} \tag{3.5}$$

The overall learning algorithm of LAE is presented in Algorithm 1. We apply a two-stage optimization to derive a generalizable forgery detector. In the first stage, we use $\mathcal{L}_1$ loss in Eq.(2.2) to learn an effective representation. In the second stage, we need the model to focus on forgery regions to learn better representations. So we exploit the active learning framework to select out challenging candidates to get their pixel-wise forgery masks. Then we reduce the learning rate one-tenth every 3 epoches and fine-tune the parameters of the encoder using the $\mathcal{L}_2$ loss in Eq.(3.5). After training the model and during the testing stage, we use latent space activation in Eq.(3) to distinguish forgery from true ones. The test images are considered to be true if $a_{i,T} > a_{i,F}$, and vice versa.

3.3 Experiments

We conduct experiments to answer the following research questions. (1) Does LAE promote the generalization accuracy when processing unseen instances, especially for those produced by alternative methods? (2) Does LAE provide better attention maps after augmenting extra supervision in the training process? (3) How do different components and hyperparameters affect the performance of LAE?

Most of the experimental setup here is same as the one we used for detecting fake images in chapter 2. The baselines methods, data, network architectures, implementation, preprocessing details are exactly same as in chapter 2. The only difference is that we now use one of the two datasets in each manipulation method (Face/Attribute/Inpainting) for training and other dataset for evaluation. Modified datasets terminology is further illustarted below.

3.3.1 Datasets

The overall empirical evaluation is performed on three types of forgery detection tasks. For each modification method(Face/Attribute/Inpainitng), we use two datasets: *source* dataset and *target* dataset. The source dataset is split into training, validation and test set, which are used to train the model, tune the hyperparameters and test the model accuracy respectively. In contrast, target dataset contains forgery images generated by an alternative method, and is only utilized to assess the true generalization ability of the detection models. Corresponding dataset statistics are given

	Face		**Attribute**		**Inpainting**	
	Face2face	FaceSwap	StarGAN	Glow	G&L	ContextAtten
Train	288000	-	41590	-	28000	-
Validation	2800	-	11952	-	6000	-
Test	2800	2800	5982	5982	6000	6000

Table 3.1: Dataset statistics for three types: face modification, attribute modification, and inpainting-based modification. For each type, source dataset is followed by target dataset. Three subsets of source dataset are used to train model, tune hyperparameter, and test model respectively. In contrast, the target dataset is only used to test the model generalization accuracy.

in Tab. 3.1. All subsets of the three modification methods are balanced, where the ratio of true and fake images are 1:1.

- **Face Modification.** This is a computer graphic based manipulation and same as the one described in section 2.1.3. Here, dataset created using Face2face [114] is considered as the source dataset, while FaceSwap [115] ones as target dataset. The process of creating these datasets is clearly explained in 2.1.3.

- **Attribute Modification.** This is a deep learning (GAN) based manipulation and same as the one decribed in section 2.1.3. Here, dataset created using StarGAN [117] is considered as the source dataset, while Glow [118] ones as target dataset. The process of creating these datasets is clearly explained in 2.1.3.

- **Inpainting-based Modification.** This is a deep learning (GAN) based manipulation and same as the one decribed in section 2.1.3. Here, dataset created using GL [119] is considered as the source dataset, while ContextAtten [120] ones as target dataset. The process of creating these datasets is clearly explained in 2.1.3.

3.3.2 Training Details

In this section, we introduce additional training details needed after first stage training, that is after getting trained AE model as mentioned in section 2.1.3 for each of the source datasets (Face2Face/StarGAN/GL).

	Face2face	StarGAN	G&L
Linear model	96.92	100.0	99.74

Table 3.2: Accuracy performance of linear model in active learning.

LAE: As mentioned in Algorithm 1, first six steps is nothing but training an AE model for each of the source dataset. So, all variants of LAE use learning rate of 0.001 and batch size of 64. For the first two hyperparameters (β_1, β_2) in Eq. 2.5, we have tuned values between 0 and 1 with 0.1 as interval and for the third (β_3), we have tried $\{0.0001, 0.001, 0.01, 0.05, 0.1, 0.5, 1.0\}$. We freeze the parameters of decoder, discriminator and only finetune encoder network parameters in the second learning stage of Algorithm 1. Note that comparator anyhow is pretrained version of VGG on ImageNet so parameters are already freezed. Just to remind, target dataset only serves testing purposes, and none of images is used to train model or tune hyperparameters. For second learning stage, learning rate is reduced by a factor of 0.1 every 3 epochs. Number of finetuning epochs depends on number of active fake images. For instance, 4 and 7 epochs work well for 100 and 500 active images respectively. During finetuning, we have tried values between 0 and 1 with 0.1 interval for λ_1, λ_2 in Eq. 3.5.

Linear model in active learning For linear model mentioned in Eq. 3.4, we use flattened output of Encoder's AvgPool2d layer as input features. Thus every image input to linear model would be represented with 512 features. We train this linear model for 5 epochs with SGD optimizer and 0.001 as learning rate. The linear model accuracy on source test sets are reported in Tab. 3.2. This indicates that the linear model could achieve similar performance with AE (or LAE without attention loss finetuning). Lastly, we finetune the encoder network on the active fake images provided by linear model using a batch size of 1 and an Adam optimizer with initial learning rate = 0.0001. For random fake active image selection (see Fig. 3.5), we collected random 50, 50, 100, 100, 100, 100 images sequentially for experiments with N=$\{50, 100, 200, 300, 400, 500\}$. This is to avoid the extra work of labeling additional attention maps.

Baselines: All baselines and their training details are same as in section 2.1.3 except that now

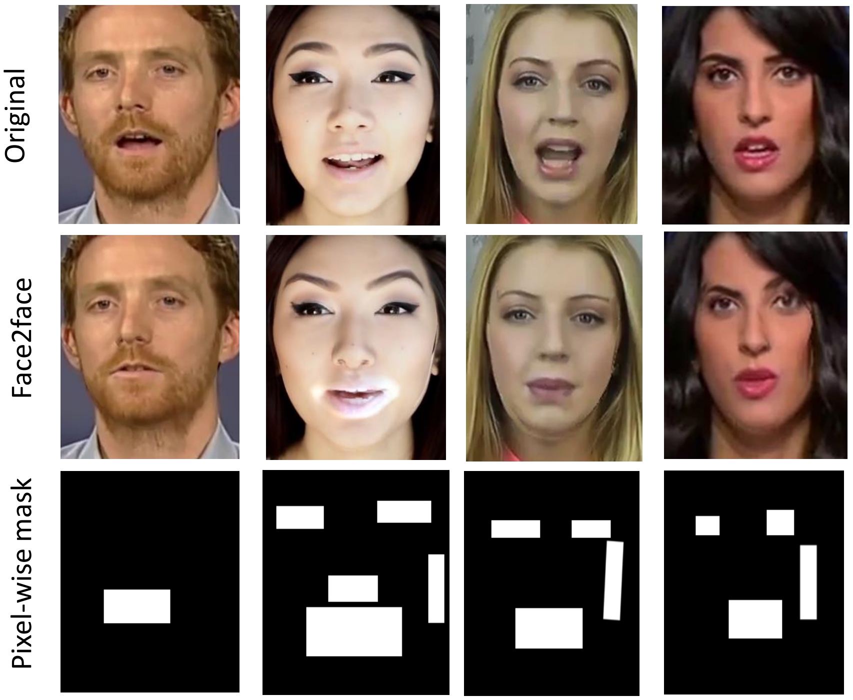


Figure 3.2: Pixel-wise ground truth masks.

train them on source datasets and test on corresponding target datasets.

Pixel-wise masks: For finetuning we need these masks to regularize explanations generated by AE (or LAE with no attention loss). Here, We illustrate some examples of Pixel-wise forgery masks in Fig. 3.2. These masks are for source Face2face generated dataset under face manipulation type of method. These pixel-wise masks give the detailed manipulated regions. With this masks, we regularize the attention loss during LAE training.

3.3.3 Generalization Accuracy Evaluation

For three manipulation methods, detection accuracy on source test set and target test set are given in Tab. 3.3. There are three interesting observations.

Generalization gap: There is a dramatic accuracy gap between source and target dataset. All baseline methods have relatively high accuracy on source test set (most of them are over 90%), while having random classification (around 50%) on target dataset. Usually the detection performance of models is calculated using the prediction accuracy on the source test set.

	Face		Attribute		Inpainting	
Models	Face2face	FaceSwap	StarGAN	Glow	G&L	ContextAtten
SuppressNet	93.86	50.92	99.98	49.94	99.08	49.98
ResidualNet	86.67	61.54	99.98	49.86	98.96	58.45
StatsNet	92.94	57.74	99.98	50.04	96.17	50.12
MesoInception	94.38	47.32	100.0	50.01	86.90	**61.34**
XceptionNet	98.02	49.94	100.0	49.67	99.86	50.16
ForensicTransfer	93.91	52.81	100.0	50.08	99.65	50.05
LAE_100	92.14	60.17	98.72	56.17	98.92	54.01
LAE_400	90.93	**63.15**	95.09	**57.01**	99.23	54.54

Table 3.3: Detection accuracy on hold-out test set of *source dataset* and generalization accuracy on test set of *target dataset*.

Due to the independent and identically distributed (i.i.d.) training-test split of data, especially in the presence of strong priors, detection model can succeed by simply recognize patterns that happen to be predictive on instances over the source test set. This is problematic, and source test set might fail to adequately measure how well detectors perform on previously unseen inputs [128]. As new types of forgery emerge quickly, it is recommended for detectors to report performance beyond hold-out test set.

LAE reduces generalization gap: LAE reduces the generalization gap by using a small ratio of extra supervision. LAE_100 and LAE_400 mean the number N_{active} is set as 100 and 400 respectively. When using 400 annotations (less than 1% than total number of training data in Tab. 3.1), we achieve state-of-the-art performance on face manipulation and attribute modification tasks. LAE outperforms best baselines by 1.61% and 6.93% respectively on target dataset of two tasks. Compared to 100 annotations, using 400 annotations has boosted the detection accuracy on target set by 2.98%, 0.84%, and 0.53% respectively. This indicates that LAE has potential to further promote generalization accuracy with more annotations. Without using any target domain fake images in the training process. Considering that new types of forgery models emerge quickly, it is crucial to guarantee the generalization performance of forensic methods.

LAE can be further improved: Despite the accuracy increase on target dataset, there is still generalization gaps. We assume that the source and target distributions should be similar for a specific forgery task. But in practice the distribution difference could be very large. The accuracy increase bound of LAE depends on the distribution difference between source and target domain. Towards this end, using a small number of target dataset data to finetune model could possibly further reduce the generalization gap, and this direction would be explored in our future research.

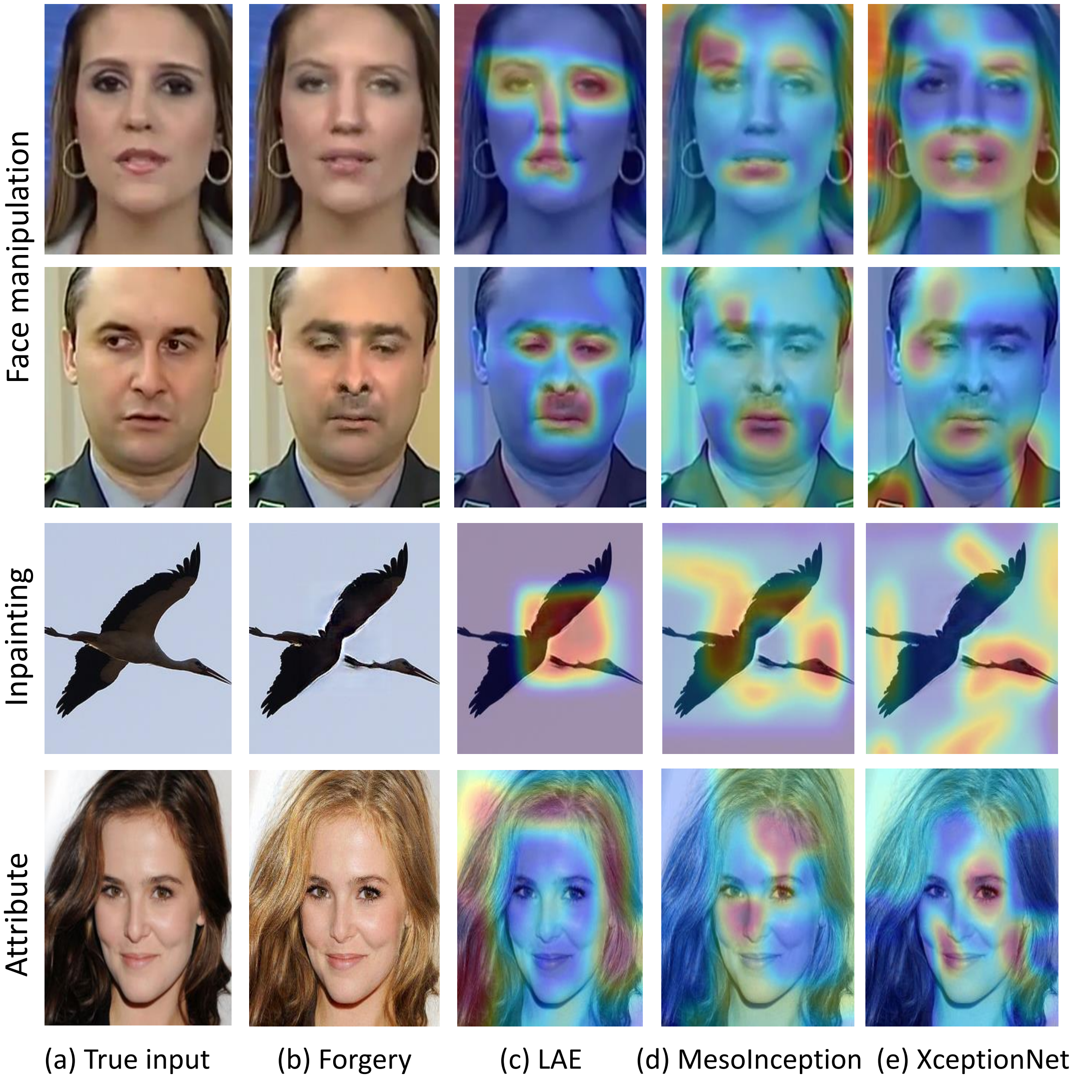
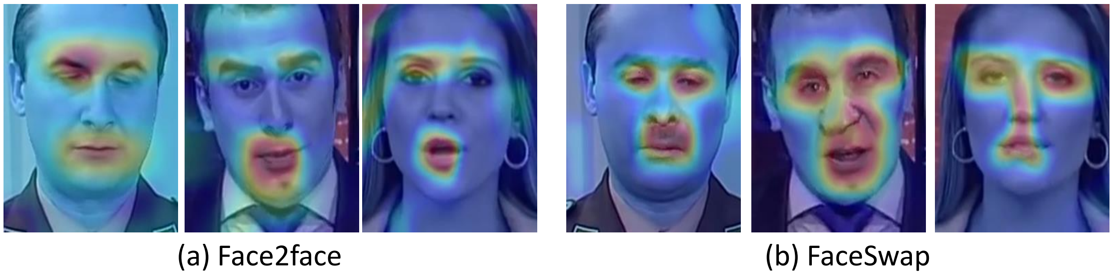


Figure 3.3: Attention map comparison with baselines.

3.3.4 Interpretability Evaluation

For all three forgery detection manipulations, we provide case studies to qualitatively illustrate the effectiveness of the generated explanation using attention maps shown from Fig. 3.3.

Comparison with baselines: LAE attention maps are compared with two baselines: MesoInception and XceptionNet, where the heatmaps for baselines are generated using Grad-CAM [129]. The visualization indicates that LAE has truly grasped the intrinsic patterns encoded in the forgery part, instead of picking up spurious and undesirable correlation during the training process. For the first two rows (face manipulation), LAE could focus attention on eyes, noses, mouths and beards. In contrast, two baselines mistakenly highlight some background region, e.g., collar and forehead. For the third and fourth row, LAE correctly focuses on the inpainted eagle neck and the modified hair region respectively. By comparison, baselines depends more on non-forgery part, e.g., wings and eyes to make detection.


(a) Face2face

(b) FaceSwap

Figure 3.4: Source and target difference via representative heatmaps.

Source and target difference: Through attention map visualizations, we observe the distribution difference of source and target dataset. For example in face manipulation detection task (see Fig. 3.4), Face2face mainly changes lips and eye brows, while FaceSwap changes mostly nose and eyes. This validates the distribution difference between source and target dataset and brings challenges to generalization accuracy.

3.3.5 Ablation and Hyperparameters Analysis

We utilize models trained on face modification data to conduct ablation and hyperparameter analysis to study the contribution of different components in LAE.

	LAE_rec	LAE_latent	LAE_latent_pixel	LAE_latent_rec	LAE
Face2face	50.39	95.82	96.57	96.92	92.14
FaceSwap	49.46	50.70	50.58	50.54	60.17

Table 3.4: Ablation analysis of LAE for face manipulation task.

Ablation analysis: We compare LAE with its ablations to identify the contributions of different components. Four ablations include: LAE_rec, trained only with reconstruction loss of Eq.(2.5); LAE_latent, using only latent space loss in Eq.(2.3); LAE_latent_pixel, using both latent space loss and pixel loss in Eq.(2.5); LAE_latent_rec, using latent space loss and whole reconstruction loss. Note that no attention loss is used in the ablations. The comparison results are given in Tab. 3.4. The results indicate that no significant differences are observed for the target FaceSwap dataset. There are several key findings. Firstly, latent space loss is the most important part, without which even source test set accuracy could drop to 50.39%. Secondly, all of pixel-wise, perceptual, and adversarial losses could contribute to performance on source test set. At the same time, no significant increase is observed on the target dataset with any combination of these losses. Thirdly, attention loss based on candidates selected via active learning could significantly increase generalization accuracy on target dataset (around 10%).

Hyperparameters analysis: We evaluate the effect of different hyperparameters towards model performance by altering the values of $\beta_1, \beta_2, \beta_3$ in Eq.(2.5) and λ_1, λ_2 in Eq.(3.5). Corresponding results are reported in Tab. 3.5 (without attention loss and active learning) and Tab. 3.6 (with attention loss and active learning) respectively. The results indicate that increase of weights for pixel

	β_1	β_2	β_3	Face2face	FaceSwap
Alter pixel	**1.0**	1.0	0.01	96.92	50.54
	0.5	1.0	0.01	96.01	50.86
	0.1	1.0	0.01	95.55	50.86
Alter perceptual	1.0	**1.0**	0.01	96.92	50.54
	1.0	**0.5**	0.01	96.74	50.82
	1.0	**0.1**	0.01	95.84	50.50
Alter adversarial	1.0	1.0	**0.1**	54.16	49.92
	1.0	1.0	**0.05**	58.28	50.01
	1.0	1.0	**0.01**	96.92	50.54

Table 3.5: Hyperparameter analysis for $\beta_1, \beta_2, \beta_3$.

	λ_1	λ_2	Face2face	FaceSwap
Fix λ_1=1.0	1.0	1.0	95.96	55.12
	1.0	0.5	96.02	52.54
	1.0	0.1	96.08	50.02
Fix λ_1=0.5	0.5	1.0	92.14	60.17
	0.5	0.5	94.48	56.31
	0.5	0.1	95.94	51.02
Fix λ_1=0.1	0.1	1.0	91.07	58.17
	0.1	0.5	92.67	53.20
	0.1	0.1	95.02	50.94

Table 3.6: Hyperparameter analysis for λ_1, λ_2.

loss and perceptual loss could enhance model performance on source test set. In contrast, a small weight for adversarial loss is beneficial for accuracy improvement. As shown in Tab. 3.5, fixing λ_1 and reducing λ_2 from 1.0 to 0.5 then to 0.1 have significantly decreased the accuracy on target dataset. This confirms the significance of attention loss in improving generalization accuracy.

Forgery ground truth number analysis: We study the effect of attention regularization by altering the number of challenging candidates(N_{active}) selected by active learning (see Fig. 3.5).

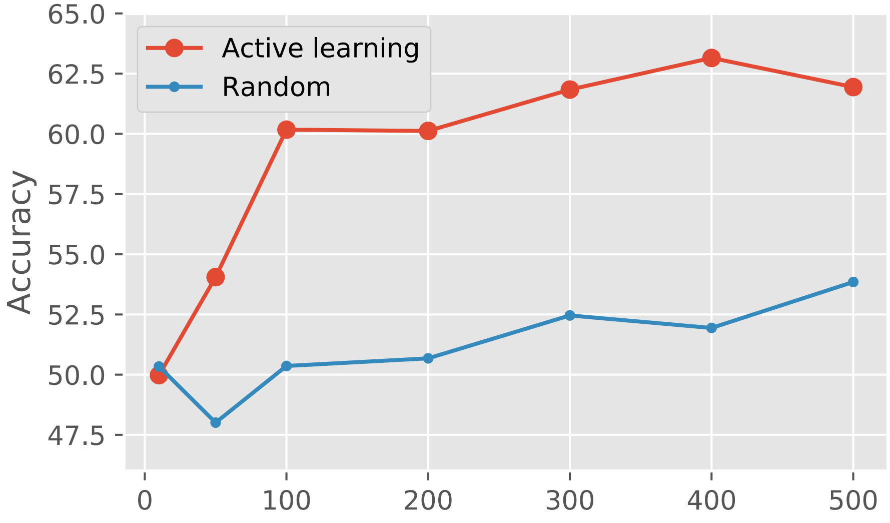


Figure 3.5: Random and active learning selection comparison. The x axis denotes annotation number which has pixel-wise masks.

There are two interesting observations. First, increasing the number of annotations typically improves model generalization, indicating the benefit of extra supervision. Second, using forgery masks for less than 0.2% of training data has increased accuracy by 10%. Considering the annotation effort of pixel-wise masks, this advantage of requiring small ratio of forgery mask annotations is significant. Some example masks are shown in Fig. 3.2.

Random vs. active learning: For challenging candidate selection, we have compared random selection with active learning based selection. The generalization result on target dataset (FaceSwap) is illustrated in Fig. 3.5. There is a dramatic gap between random selection and active learning. For instance, active learning could increase target dataset accuracy by 9.81% when the annotation number is 100 ($< 0.2\%$ of training data). This indicates that active learning is effective in terms of selecting challenging candidates.

3.4 More Interpretation Visualizations

In this section, we provide more heatmap visualizations to better understand the effectiveness of attention loss finetuning and active learning.

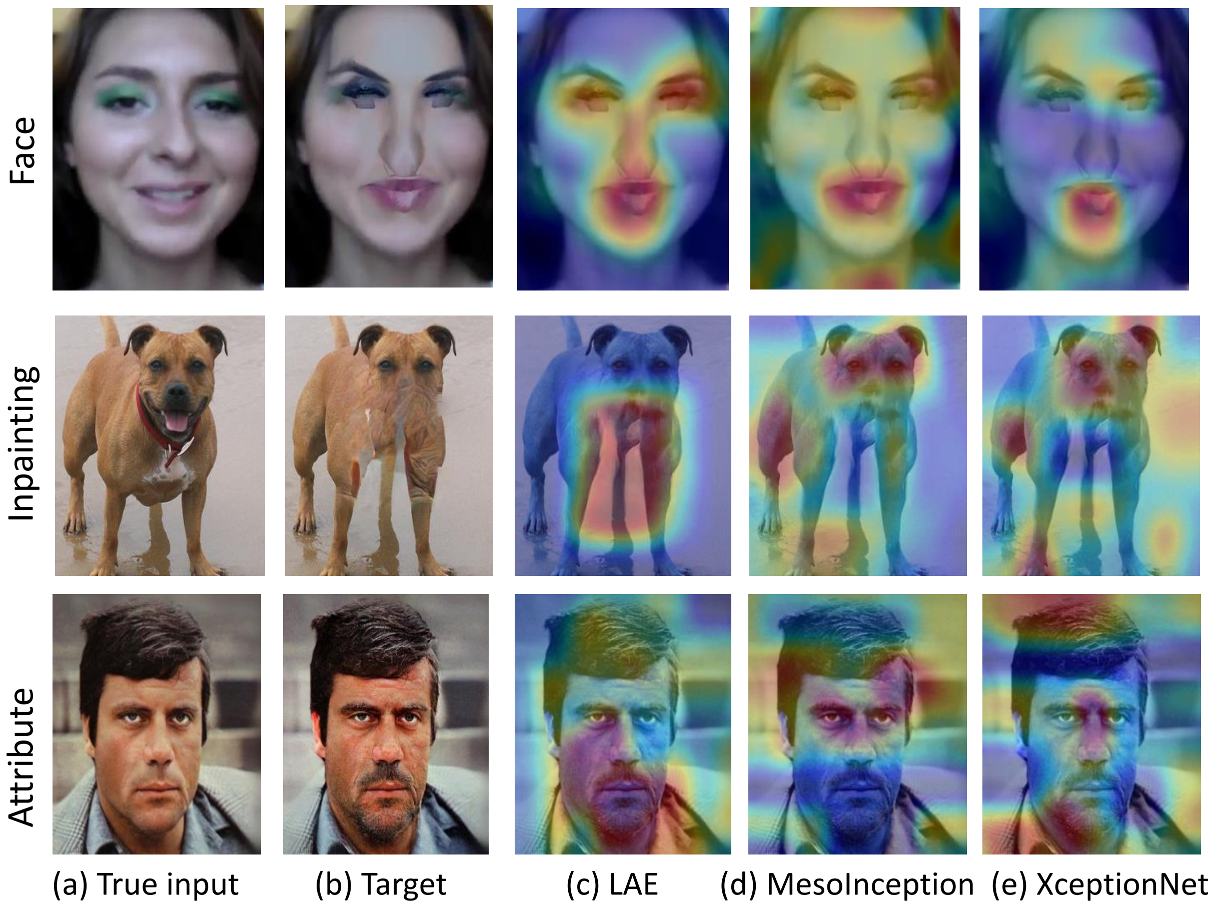


Figure 3.6: Attention map comparison with baselines.

Visualization comparisons: We provide heatmap visualization in Fig. 3.6. For three tasks, we compare LAE heatmaps with two baselines. These visualizations validate that LAE makes detection based on right and justified reasons.

Effectiveness of attention loss: To qualitatively evaluate the effectiveness of attention loss and active learning, we provide ablation visualizations in Fig. 3.7. Specifically, we compare LAE_no_atten (without using attention loss and active learning) and LAE. Before using attention loss, we can observe that the model does not accurately rely on the forgery region to make decisions. After finetuning with attention loss with a small ratio of samples provided by active learning, the model learns to concentrate on the forgery part to make detection.

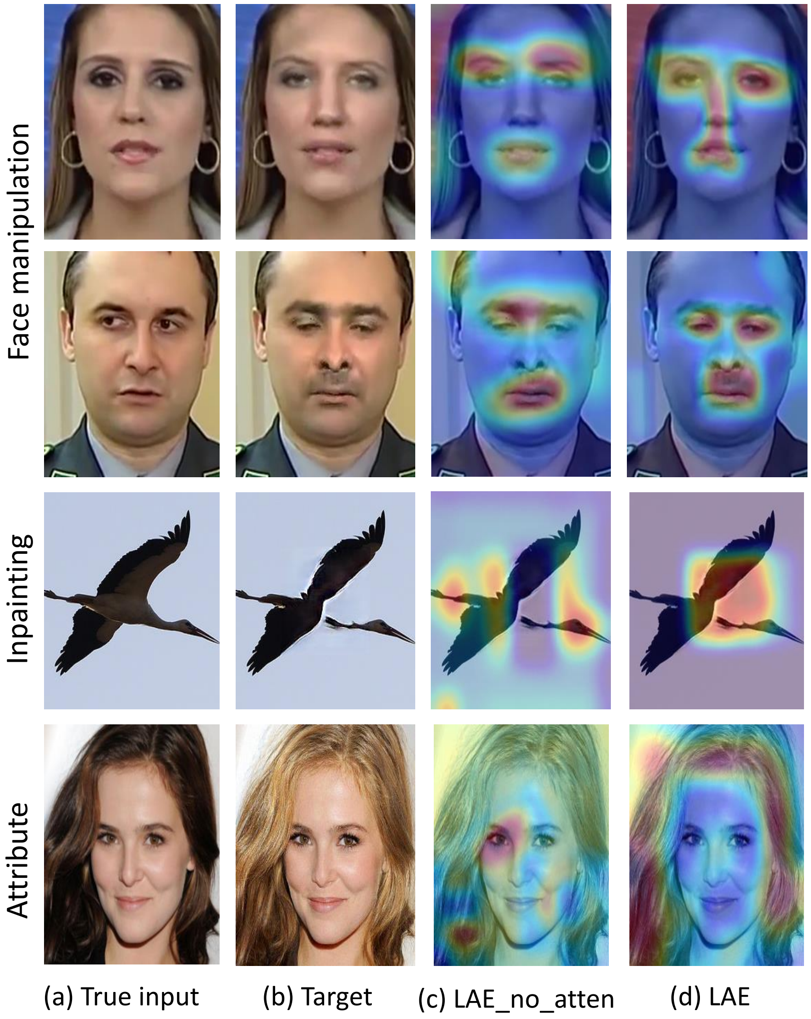


Figure 3.7: Effectiveness of attention loss.

3.5 Discussion

From all these experiments, it's clear that explanations generated from a task can be helpful in further improving the learning of that task. In our case, Locality-aware AutoEncoder (LAE) has a higher probability to look at forgery region rather than unwanted bias and artifacts to make predictions. Empirical analysis further demonstrates that LAE has superior generalization performance on data generated by alternative forgery methods that are related to the source forgery method. This boost in generalization comes by making predictions relying on correct forgery evidence.

In addition, our proposed active learning framework also found to be extremely useful in significantly reducing the efforts to get forgery masks (less than 1% of training data).

However, due to the inherent difficulty of the detection problem, we still could observe generalization gap between source dataset and target dataset that is generated by alternative methods. Although they are related and belong to the same task/modification method, there still remains slight distribution differences between them. Using transfer learning and other techniques to further reduce this generalization gap could be explored in our future research.

4. EXPLAINING LANGUAGE BASED FAKE NEWS

In this section, we first build fake text detection models mostly with deep learning methods and later explain their decisions. For a given news statement, we want our models to correctly predict if this news statement is real or fake and also provide the explanation why the decision is made. Overall, we aim to make neural networks for NLP use cases more interpretable without sacrificing much on accuracy.

4.1 Detecting Fake Text

Fake text detection is one of text classification tasks that takes a sentence or paragraph as input, and determines if it is real or fake. There is no state-of-the-art dataset for this task but many of the available datasets are crawled from either Politifact [5] or Snopes [1] which are basically fact-checking websites where a group of journalists labels a news trending on web as either true or fake. More info about datasets is illustrated in section 4.1.2.

4.1.1 Methodology

As our main focus is on explaining fake text detection, we came up with two traditional deep learning methods used for text classification - LSTM-based, CNN-based and one random forest based shallow model which uses predefined linguistic features in the input layer. Below we illustrate them in further detail.

ATTN: This is a CNN, attention based architecture which takes a sentence as input and outputs softmax scores for two labels. This is designed to analyze text simply from semantic perspective. For better semantic analysis, we employ several techniques, including pre-trained word embeddings, convolutional neural network, and self-attention mechanism. Self-attention is used because it can capture global relationships between different words efficiently [130]. Overall architecture is straightforward and contains input embedding layer followed by several layers of 1D convolutionals, self-attention layers, maxpooling and lastly a softmax layer.

MIMIC Teacher: Mimic learning is also called knowledge distillation approach which approximates a teacher model (usually deep network) with a student model(usually shallow network) [105]. More info about MIMIC student is provided in section 4.1.2. Here let's restrict our focus to teacher network which takes a news sentence and it's attributes like speaker, subject of the news etc. and outputs softmax scores for two labels. Architecture includes an embedding layer followed by CNN with maxpool to capture sentence representation, parallel Bi-LSTM modules to extract features from rest of the attributes [69]. All features are concatenated and passed through fully connected layer followed by a softmax layer.

PERT: It is designed for news statement analysis from linguistic features perspective. Architecture inludes a feature engineering step followed by a XGBoost classifier which is an optimized gradient boosting algorithm that works by parallel processing, tree-pruning and regularization. For effective feature engineering step, we employ eight linguistic features, including *Adjective ratio*, *Noun ratio*, *Verb ratio*, *Preposition ratio*, *Sentiment score*, *Normalized text length*, *Whether contains the mark "?"*, *Whether contains the mark "!"*. For each news sentence input, we extract its linguistic features and train an XGBoost classifier using these features. The trained XGBoost is then used to make predictions for new items.

4.1.2 Experiments

In this section, we introduce the overall experimental setups including datasets, baselines, networks architecture and training details.

Dataset: Many of the datasets related to deceptive customer reviews detection are crowdsourced datasets[131, 132]. However, these are not suitable for fake news detection as the positive training data in them are collected from a simulated environment and also fake news on social media are generally much shorter compared to customer reviews. Later, there has been efforts [133, 134] to construct fake news dataset from fact checking websites [135, 5], but however they are small in size (less than 350 samples). Recently, LIAR dataset [69] with 12.8K samples was introduced to facilitate the development of deep learning methods for this task. This dataset is crawled from Politifact website [5], which covers a widerange of political topics with fine-grained labels. This is a very good dataset for detection task however this data do not includes some impor-

Dataset	Train	Val	Test
Politifact	4083	510	511

Table 4.1: Fake news detection dataset statistics crawled from Politifact [5]. All splits have 1:1 true and false labeled examples.

tant attributes that might make detection as fake. For example, whom the speaker is targetting in the news, what is the topic/subject of the news (taxes, crime etc.) and more importantly this dataset contains more than 5 years old news which are not super useful to our use case and can also cause problems during human studies (lack of awareness about that old topic etc.). So all these reasons motivated us to create another dataset using the same website but this time data is collected with more attributes and also most recent ones. Our crawled dataset is further illustarted below.

Politifact The news data we crawled comes from a political fact-checking website, named PolitiFact [5]. It is a Pulitzer prize-winning website containing tons of political news with diversified categories. The reasons why we employ this data source are in four folds. First, this website was used by most of the previously released datasets for fake news. Second, PolitiFact provides professional justification and fine-grained labels for all news items, where the core principles in independence, transparency and fairness guarantee its high credibility among the public. Third, the news collected by PolitiFact have various attribute information, which directly meets our data requirement for analysis. Fourth, raw data in PolitiFact has an API [1], and it is convenient to obtain the customized dataset.

Data Preprocessing When crawling and processing the data, we only keep the attributes which are highly related to news fakeness. Specifically, the maintained attributes include *Subject*, *Context*, *Speaker*, *Targeting* and *Statement*, although some news items may not have all five attributes. Besides, to effectively measure the fakeness of news and train the system, we transform the original multi-class data to binary data where each news item is labelled as either *True* or *False*[2]. In particular, labels with *Mostly True*, *Half True*, *No Flip*, *Half Flip* are switched to the positive label

[1] http://static.politifact.com.s3.amazonaws.com/api/v2apidoc.html

[2] A news item labelled as False is regarded as the fake news.

Instance	Label
Statement: It is unusual for a White House official like former National Security Adviser Susan Rice to make unmasking requests. **Speaker:** Tom Cotton **Context:** an interview on CNN **Targetting:** Susan Rice **Subject:** Foreign Policy, Homeland Security, Privacy	False
Statement: Says Ted Cruz distributed the ad showing a nude Melania Trump on a rug. **Speaker:** Donald Trump **Context:** an interview on CNN **Targetting:** Ted Cruz **Subject:** Campaign Finance, Candidate Biography, Elections, Legal Issues, Negative Campaigning, Pop Culture.	False
Statement: Obama’s secretary of energy, Dr. Steven Chu, ’has said publicly he wants us to pay European levels (for gasoline), and that would be USD9 or USD10 a gallon.’ **Speaker:** Newt Gingrich **Context:** an appearance on "Fox News Sunday" **Targetting:** Steven Chu **Subject:** Gas Prices	True

Table 4.2: Few examples from our dataset crawled from Politifact [5].

True, and labels with *Mostly False*, *Pants On Fire*, *Full Flop* are switched to the negative label *False*. Instead of using multiple discrete labels, we use the final prediction scores to indicate the level of fakeness, where higher scores correspond to higher fakeness level. Overall data statistics are shown in are shown in Tab. 4.1. All splits are balanced in terms of true and false labeled examples. Some examples from the dataset are shown in Tab. 4.2.

Baselines: Although our main focus is not detection accuracy, we still want to compare our above defined three models with few baselines so that we are not compromising too much on accuracy for the sake for better explainability. All baselines operate only on news statement and do not use any other attribute information in their input (same as ATTN, PERT).

We have considered five baselines: Naive Bayes (NB), a regularized logistic regression (LR), SVM [136], Bi-LSTM [137, 138], CNN [139]. For NB, LR and SVM, we have used bag-of-words representation with two type of input representations - count based, TF-IDF which are further illustrated in the training details section along with hyper parameter, word embeddings info of deep learning models.

Training details: For fair comparison, all models are trained on the same data and tested on the same test set. For deep learning models, we have used early stopping strategy using validation loss as stopping criteria with patience set to 5 and for a maximum of 20 epochs. For non-deep learning models, we used grid search to tune the hyperparameters and later reported results on the same test set as being used by deep learning methods. For training of deep learning models, we use the Adam optimizer [127] with a learning rate of 0.001, batchsize of 64 and default values for the moments $\beta 1$ = 0.9, $\beta 2$ = 0.999 and epsilon set to 10^{-8}. Model specific training details are illustrated below. On the other hand, Stochastic gradient descent is used for non-deep learning model learning process.

- **ATTN.** We employ pretrained 300-dimensional word2vec embeddings from Google News [140] and thus each vector representation has a dimension of 300 (i.e. E$= 300$ in Fig. 4.1). Each spatial location learns a 512-dimensional vector representation for each word (i.e. D$=$ 512 in Fig. 4.1). For 1D convolutional part, kernel size is set to 1. Used Tensorflow for implementation.

- **MIMIC Teacher.** We use Glove Wikipedia 6B word embeddings [141] and BiLSTM with 128 hidden units. Experimented with both CNN and LSTM layers for statement part of the input and found that CNN with (2,3,4) filter sizes, 128 filters give better results. For rest of the attributes, we use BiLSTM module. Used Keras for implementation.

- **PERT.** We have used NLTK python module for feature engineering part to get our required 8 linguistic features. Used grid search to tune two important hyperparameters - max depth of the trees, eta (which is analogous to learning rate). Used Keras for implementation.

  Baselines: For Naive Bayes (NB), LR, SVM, we use the bag-of-words (BoW) input representation by selecting 1000 most frequent words from the training set. For each model, we ran two set of experiments - one using counts, other using TF-IDF scores. In the results, we reported best of the two. In count based representation, we use counts of each word as the features. For TF-IDF (term-frequency inverse-document-frequency) version, we use the counts as the term-frequency. The inverse document frequency is the logarithm of the division between total number of samples and number of samples with the word in the training set. The features are normalized by dividing the largest feature value. For BiLSTM model, experimented with 64, 128 hidden units and found 128 gave better results. For CNN, filter sizes of (2,3,4) with 128 filters gave better results.

Accuracy Evaluation: All models are evaluated on the same test set and accuracy is used as metric to compare them. Tab. 4.3 shows the comparison between our three models and several baselines. Results highlight that usage of additional attribute information in the input can give you better performance as MIMIC teacher outperforms all other models which indeed just use news statement. Poor performance of PERT can be because of using only 8 linguistic features to make prediction. However, our method is flexible to incorporate additional fetaures.

For NB, LR, SVM, we also tried increasing the number of frequent words for BoW representation from 1000 to 5000 and haven't found much significant improvements. Although ATT has

Models	**Test set accuracy**
Naive Bayes	56.87 %
LR	54.67 %
SVM	57.92 %
Bi-LSTM	66.34 %
CNN	67.96 %
ATT	67.3 %
MIMIC Teacher	**68.98** %
PERT	53.2 %

Table 4.3: Detection accuracy of several models for our Politifact dataset.

less accuracy than traditional CNN architecture, but it has a transparent architecture which can self explain its predictions using word or phrase importance which is further illustrated in section 4.2.1. The better performance of CNN over Bi-LSTM on this dataset might be because of short sentences and identifying some simple features like angry terms, sadness, abuses, named entities that trigger the sentiment might be more useful for prediction rather than learning long-range semantic dependencies with RNNs.

By comparing our three models with baselines, we can conclude that we are not sacrifising much on accuracy (except PERT model) in the process on building more explainable models. Although traditional CNN performs better than some of our methods (ATT, PERT), but that is still a black-box model unline ATT, PERT. Details in this regard are further illustrated in section 4.2.1.

4.2 Explaining the Detections

In this section, we provide three different types of explanations for the above detection task - word/phrase importance, attribute importance, linguistic feature importance. We will explain the same detection models as in section 4.1 using post-hoc (ex: mimic learning, perturbation) and intrinsic (ex: attention) type explanation methods.

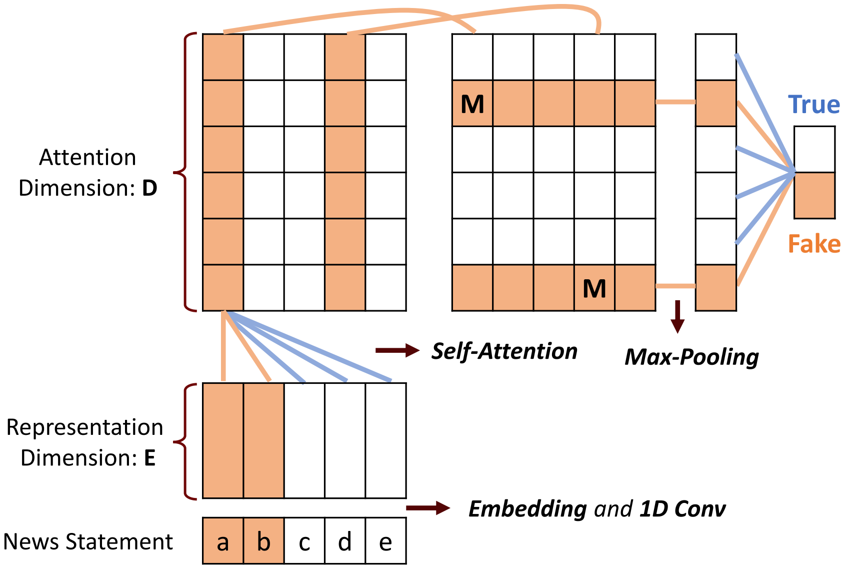


Figure 4.1: Generating word/phrase importance from ATTN framework.

4.2.1 Word/Phrase Importance

These could be generated from our ATTN model in section 4.1.1. It is indeed a self-explanatory model that incorporate explainability directly into its structure because of attention layers. Thus it falls under intrinsic explainability category. It is called self-explanatory for following reasons: first, it will generate a weight matrix based on its input only so that the interpretation results are input-dependent. Second, the output is a weighted sum of input vectors based on the whole input, where the weighted matrix is generated using the input itself. In this way, when we try to answer which spatial locations in the previous layer contribute most to a certain location of the next layer, all spatial locations are considered since the receptive field of the self-attention layer is the whole input. It is different from convolutional layers where the receptive field is based on the kernel size, which is much smaller than the input length in most case. After the self-attention layers, a max-pooling layer and a final fully-connected layer are employed to produce a final prediction. After prediction, we perform a backtracking from the output prediction to the input sentence to investigate which input words contribute most to the output decision as shown in Fig. 4.1. To show how our model is interpretable, let's walk through an example. Assume we have a sentence with

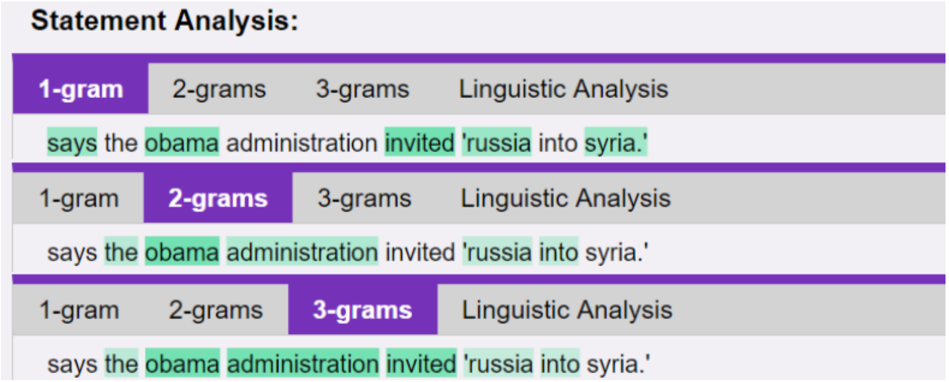
5 words. after the embedding layer and 1D convolutional layers, the size becomes (E, 5). Then it passes through attention layers and the size of the matrix becomes (6, 5) where we assume the dimension of self-attention model is 6. Next, we obtain a (6, 1) vector using max-pooling which only keeps the maximum value for each feature. Finally, the fully-connected layer with softmax activation leads to the prediction (lets say class is either 1 or 2).

If the prediction of this input sentence is class 2, we can check the weights of the fully-connected layer which connect with location 2 of output and find the k highest weights (shown in red, and here k=2). Then we know the second row and sixth row contribute most to the final prediction. In these two rows, we can find the max value in each row and the corresponding columns (column 1 and 4 in this example), which means the features at these two spatial locations contribute most. Note that each row in Fig 4.1 corresponds to one type of feature while each column represents a spatial location. Next, we can check the weight matrix generated by attention layer to see which columns in the previous layer contribute most to these two columns. Repeat such back-tracking to the output of 1D convolutional layers, we know which columns of the output of 1D convolutional layers contribute most to the final decision. Since we set the kernel size and stride equal to 1, each column of output of convolutional layers corresponds to one input word. Then we know how different words affect the final decision. Similarly, if we wish to study two-grams or three-grams, we can simply set the kernel size to two and three, respectively.

Lastly, we have shown an example for this type of explanation in Fig. 4.2 which is generated from our trained ATTN model. We can see that model captured those important nouns (obama, russia) and verbs (says, invited) in the sentence for its prediction.

4.2.2 Attribute Importance

For fake images, just looking at the image we can guess possible modified artifacts in the image. Unlike fake images, fake part in the language based (text) news is difficult to understand just from news statement. There is additional domain knowledge involved to do proper fact checking. Some examples of this knowledge can be source of the news, context of the news (a twitter post, a tv interview etc.), speaker of the statement etc. Fortunately, our dataset has all that additional at-



Statement Analysis:

says the obama administration invited 'russia into syria.'

says the obama administration invited 'russia into syria.'

says the obama administration invited 'russia into syria.'

Figure 4.2: Example explanation from ATTN model for a given news statement. one-gram importances in the top and two-gram in the middle and three-gram importances in the bottom of the figure.

tributes and we also have trained a MIMIC teacher model using those attributes as input. So in this section, we will explain the predictions of that teacher model in the form of attribute importance for fake news classification task.

MIMIC Student: We use knowledge distillation approach (also called mimic learning) to approximate the teacher model (deep architecture) trained in section 4.1.2 with a student model (random forest or more specifically XGBOOST). Basically, the overall idea of MIMIC framework is to mimic the performance of deep neural networks with the shallow models(generally tree ensemble models) so that we can keep the good performance from neural networks and good explainability from shallow modles. Later, use the student model to understand the attribute importance in classifying a news as True/Fake. With this MIMIC framework, we can achieve both model-level and instance-level interpretations, and further obtain relevant supporting examples from training data. By model-level, we mean feature importance over all trained examples. On the other hand, instance-level just means feature importance for a specific instance prediction. The structure of MIMIC is illustrated in Figure 4.3.

As we have already trained a MIMIC teacher model, we now obtain the soft labels from the teacher model and further use them to train XGBOOST, a shallow and interpretable method. The overall architecture of this framework is shown in Fig. 4.3. We use the same training setup as described in section 4.1.2. It's final performance is similar to the teacher model as indicated in

Table. Importance hyperparameters for student model are number of decision trees and in our case 80 trees gave best results on validation set. Also note that, during inference, we do not use teacher model anymore, input is directly passed through student model.

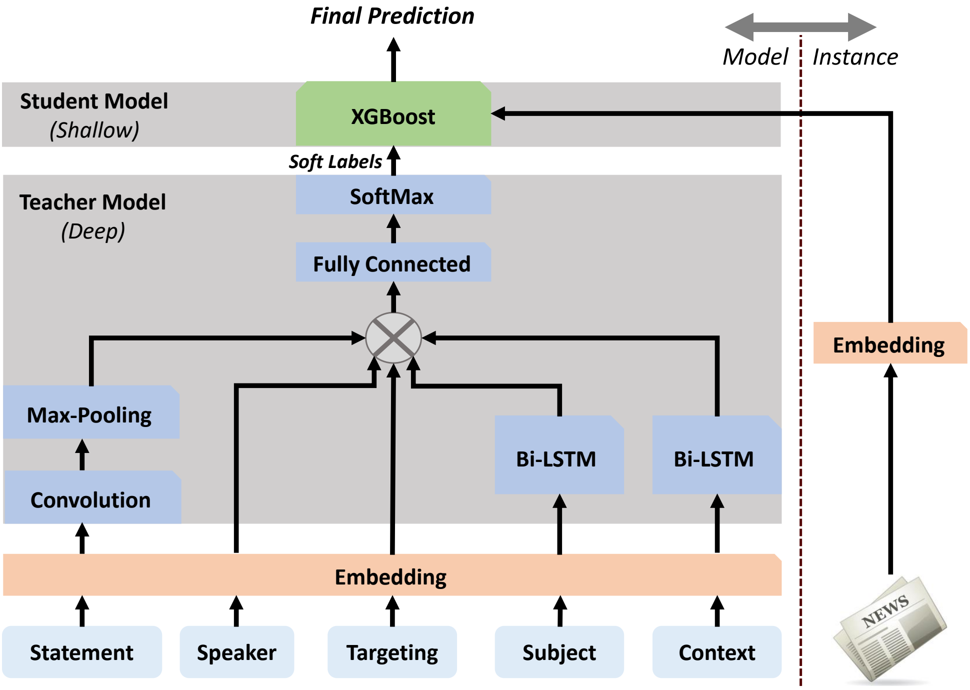


Figure 4.3: The structure of MIMIC framework.

Explanations from MIMIC: As we want to interpret fake news from the attribute perspective, in our scenario, each news item contains 5 different attributes, i.e. Subject, Speaker, Context, Targeting and Statement. Using MIMIC framework, we want to quantify the relevant contributions of these four attributes. For example, some of news items can be fake due to their Subject attribute, and some of news can be true due to their Speaker or Statement attribute. As during inference, input is directly passed through student model, we can collect attribute importance of fake news input by analysing relative node importance in decision trees. Thus, it's straightforward to generate explanations from a forest based model for each instance prediction [142, 143, 144].

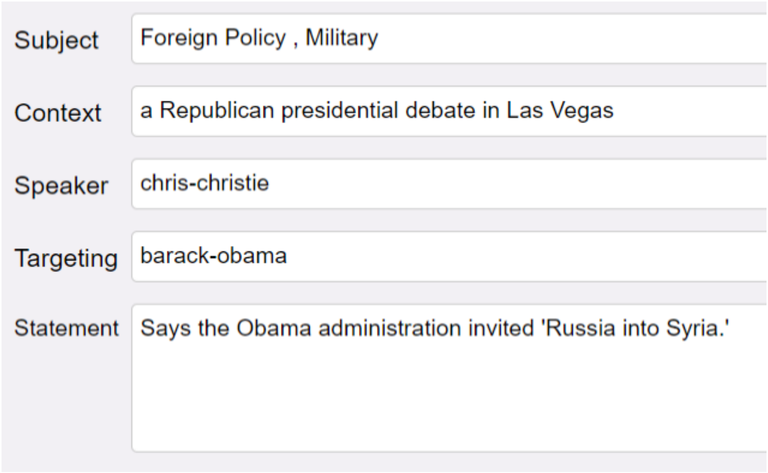
Attribute contribution scores These contribution scores for each input news instance are obtained by finding the relative node importance of each attribute in the decision trees. For that we employed some simple heuristics like the number of times an attribute is used to make a decision at intermediate nodes across all trees, the average gain of the attribute when it is used in trees. Finally, we normalize all five attribute scores to make them sum to 1.0.

Activated path with attribute nodes As trained XGBOOST has many decision trees, only few paths in them would be activated during inference stage. We collect those paths and sort them based on their leaf node scores. For example Fig. 4.4b shown an activated path for the input instance shown in Fig. 4.4a. Although these paths are a bit complex, we could still get some idea of what happened inside the student model, like in this example, we can see news context is involved in initial analysis followed by reviewing actual statement content and at the end some speaker-based decision making nodes. *Supporting examples* For each instance prediction, we employed a rule based approach to get five other examples from the training data whose attribute importance order is same as the current instance attribute score order, same prediction label as the current one, with some common words in top two attribute input fields. However, this looks very naive and could be improved with other Bayesian approaches like [145, 146], which could be a promising work for future.

4.2.3 Linguistic Feature Importance

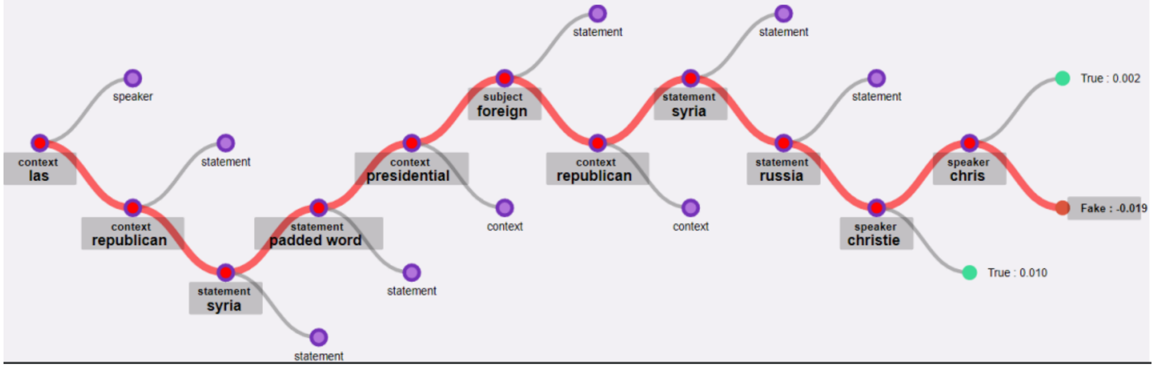
In this section, we would like to explain a fake news detection using interpretable linguistic features. Hence we utilize the trained PERT model in section 4.1.1 and use perturbation-based method on it to provide the required explanations. As input to our model is eight linguistic features (after feature engineering step), we utilize a perturbation based explanation method to get importance of each of these eight features.

The idea of perturbation method to get feature importance is that feature importance can be measured by observing how much the score (such as accuracy, etc.) decreases when a feature is not available. To this end, we can remove a feature from the dataset, and then re-train the classifier and check the changes of score. Since re-training is computationally expensive, we replace



Subject	Foreign Policy , Military
Context	a Republican presidential debate in Las Vegas
Speaker	chris-christie
Targeting	barack-obama
Statement	Says the Obama administration invited 'Russia into Syria.'

(a) Input to the student model.



(b) Activated path of a decision tree in student's forest.

Figure 4.4: Example explanation from MIMIC framework.

the feature value with random noise, drawn from the same distribution as the original one, instead of removing. The computed prediction difference is utilized as the significance score for the corresponding feature. All this procedure is illustrated in Fig. 4.5.

An example prediction of PERT model is shown in Fig. 4.6 which indicate that for that instance, model mainly focuses on nouns, verbs, prepostions in the sentence to make its decision.

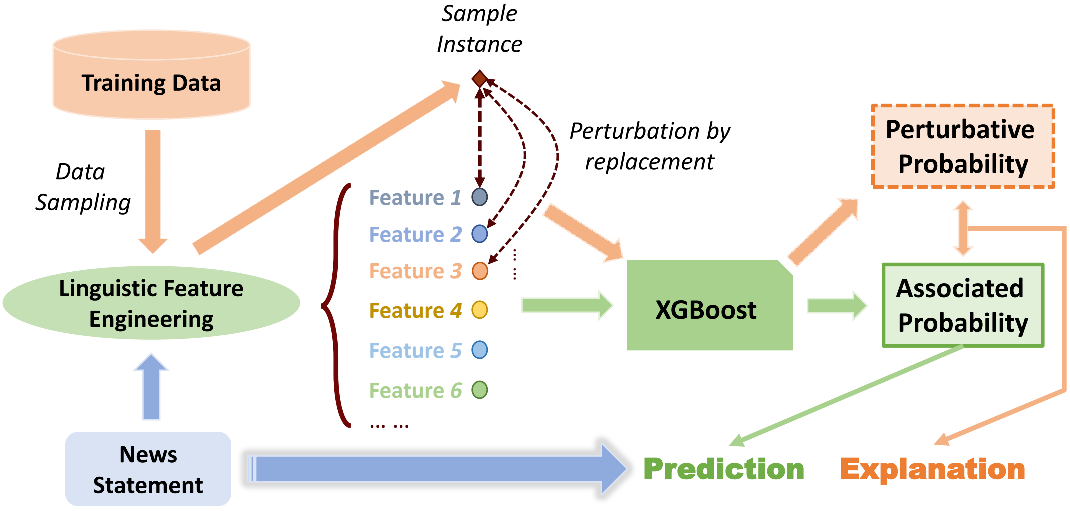


Figure 4.5: Explanations from PERT model using perturbation.

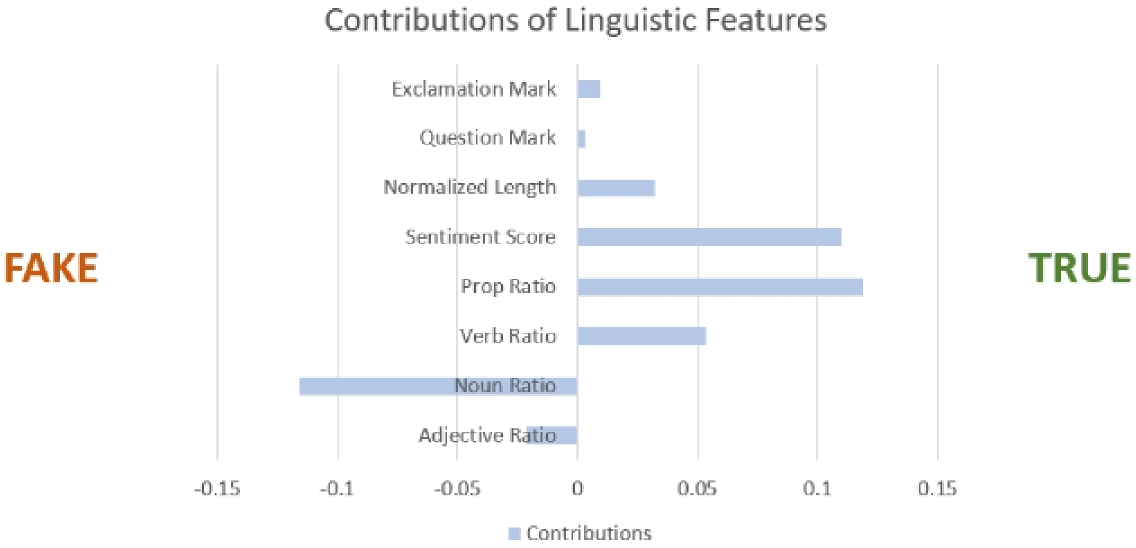


Figure 4.6: PERT explanation for the news "*The senate seat won in a special election by a democrat had been held for more than thirty years by republicans.*" Blue bars indicate the contribution score of each linguistic feature and bars pointing to right indicate that they are positively contributed for true class prediction and pointing to left indicate positive contribution for fake prediction. PERT confidence scores for this instance are 0.58 for true and 0.42 for fake.

5. INTERACTIVE SYSTEM TO EXPLAIN FAKE NEWS

We designed an explainable fake news detector, named XFake [147], to help end-users identify the news credibility. This is implemented as a web application where users can query a news and get instant prediction along with an explanation to help them understand why the system thinks so. Explanations could be from multiple perspectives like attribute importance, word/phrase significance, linguistic features as well as relevant supporting examples.

5.1 System Design

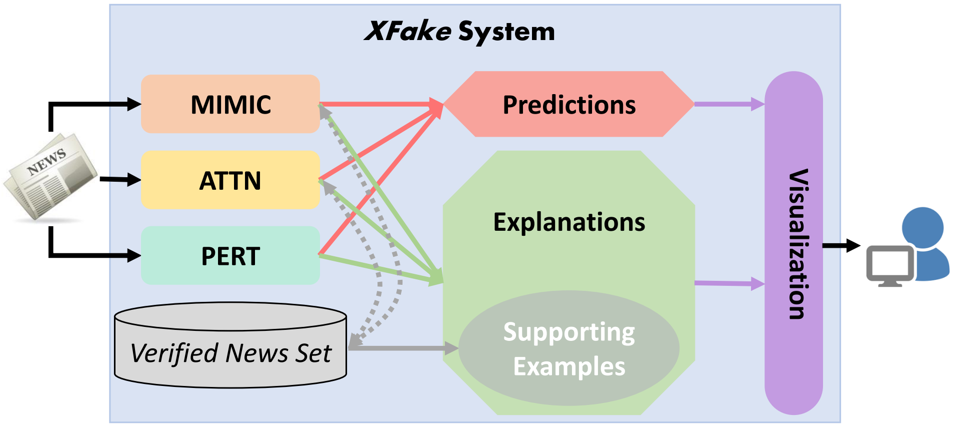


Figure 5.1: The architecture of XFake system.

Here we given an overview of different components of the system to better understand input and output. We utilize the detection/explanation methods (*MIMIC*, *ATTN* and *PERT*) developed for language based fake news in section 4.1.1. Visualization and API call details are explained in detail under section 5.2 and 5.3.

Input: We can incorporate multiple information about the news like statement (ex: He is mathematically out of winning the race), speaker (ex: Donald Trump), subject (ex: Elections), context (ex: a tweet), targetting (Ted Cruz). It is not mandatory to provide all five attributes but we expect users to provide statement attribute.

Output: There are two possible output types for each instance input. 1) Final prediction of the news or classification result, 2) Corresponding explanations for that prediction.

- **Predictions.** This is the classification output result with "True" and "Fake" labels. Classification score for each label is also included as shown in Fig. 5.2. As we get prediction output from three methods, we take weighted summation of all three prediction scores to get an unified score indicating the probability for the given input to be a fake news i.e the higher the prediction score, more likely that it is a fake news. Here weights are assigned based on performance of individual models on test set described in 4.1.2 which came out to be *0.36*, *0.36*, *0.28* respectively for *MIMIC*, *ATTN* and *PERT*.

- **Explanations.** *MIMIC*, *ATTN* and *PERT* analyze news items from different perspectives. Details about how each of these methods generate explanation are provided in section 4.2. Overall, we would have 4 types of explanations as described below.

  Attribute Analysis: This is obtained from *MIMIC* framework which basically tells the importance of attribute in the form of a score between 0 and 1 for each attribute. 1 indicate highly important.

  Statement Analysis: These come from *ATT, PERT* model based on just analyzing input statement from semantic and syntactic perspectives respectively. Thus we output importance for 1-grams, 2-grams, 3-grams in the sentence. Also, we output linguistic feature importance like (Noun ratio, Verbs ratio, length of sentence etc.) in the form scores between -1 and 1 for each feature. positive values indicate it's contribution is aligned with the final prediction (True/Fake). High score indicate it's important.

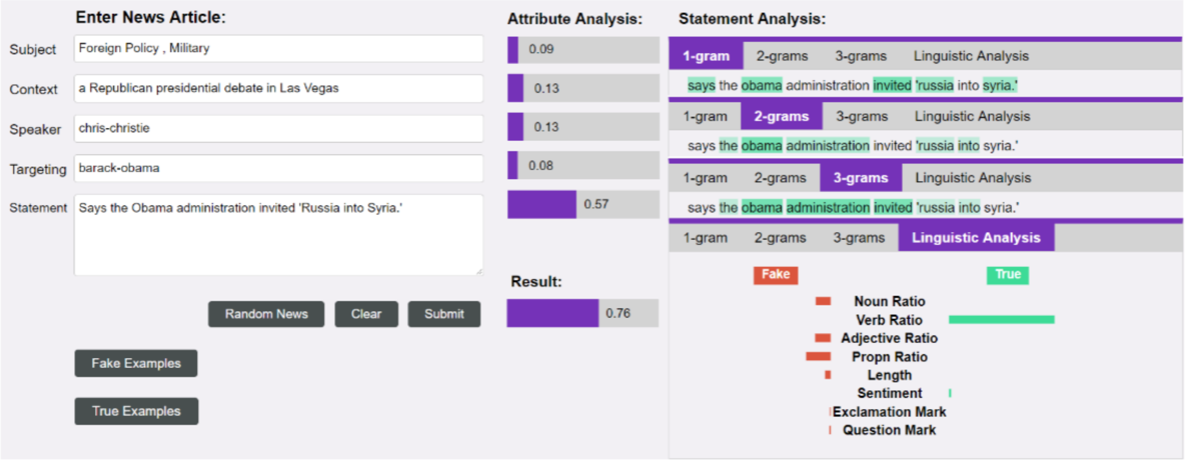
  Supporting Examples: This gives top five new articles related to the current input and helps the user to understand the decision making by providing evidence of similar news. We assign different similarity scores for different supporting examples based on the matching extent between the input and support. Supporting examples with higher scores would be considered more informative in delivering explanation. These examples are generated from *MIMIC*, *ATTN* by retrieving samples from original dataset using the current prediction explanations (i.e. important attributes or important words/phrases). More details about how they are being generated are explained in section 4.2. Showing these samples which are highly similar (in terms of prediction label, common words, attribute and n-grams score) to the input news would be helpful for users to understand the working patterns of XFake.

Decision paths: We show the important trees in the pre-trained *MIMIC* model and also activated paths in them for the current instance. These activated paths in the decision tree can tell us a naive idea of attribute order used for the prediction. End-user is able to review all tree models used in the system. We show (red path) the decision paths that help user to understand which nodes activations resulted in the decision making. Fig. 5.3 shows our interactive tree, activated path visualization. More about interactive visualizations is further explored in section 5.2.

We will explore components used on client and server side in the next sections.

5.2 Client/Front-end

This development is done using HTML, CSS, JavaScript. We visualize both prediction and explanations by *D3 JavaScript*. Visualization mainly lies in three aspects. First, for numerical values, such as prediction score and attribute significance, we visualize them by histograms, which straightforwardly indicate the results and influences. Second, to enhance the explanability for word/phrase attribution, we visualize the outputs by highlighting important words/phrases with heatmaps, where the darkness positively relates to the importance of word/phrase. Third, for better model explanability, the ensemble trees are visualized with interactive diagrams which are capable of showing both overall structure and specific activated paths (depending on the input). Through those visualization schemes, users could have a better sense towards XFake about why certain news are classified as fake or true.



Subject: Foreign Policy , Military

Context: a Republican presidential debate in Las Vegas

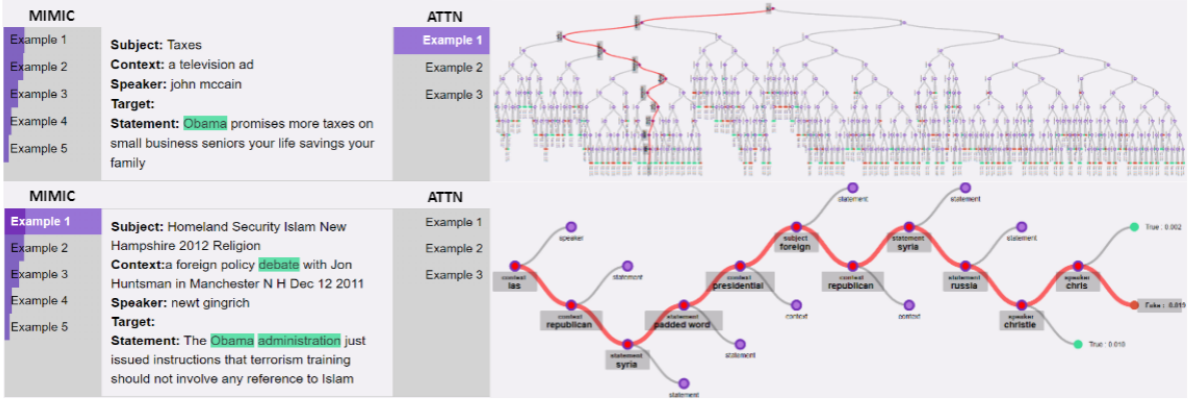
Speaker: chris-christie

Targeting: barack-obama

Statement: Says the Obama administration invited 'Russia into Syria.'

says the obama administration invited 'russia into syria.'

Figure 5.2: Prediction and explanation from XFake.



Subject: Taxes
Context: a television ad
Speaker: john mccain
Target:
Statement: Obama promises more taxes on small business seniors your life savings your family

Subject: Homeland Security Islam New Hampshire 2012 Religion
Context:a foreign policy debate with Jon Huntsman in Manchester N H Dec 12 2011
Speaker: newt gingrich
Target:
Statement: The Obama administration just issued instructions that terrorism training should not involve any reference to Islam

Figure 5.3: Supporting examples and ensemble trees from XFake.

Considering the fake news identification scenario, we show a specific case demonstration of XFake as follows.

Demonstration: As shown in Fig. 5.2, users can input news into the text boxes. We also provide a button "Random News" to help users explore the system, which is used to retrieve random items from our test set. Similarly, buttons "Fake Examples" and "True Examples" are also provided to help quickly access some representative fake and true news. After clicking "Submit", users would obtain all the outputs including both prediction and explanation in a few seconds. As for the prediction of the example in Figure 5, we get the score 0.76, which means that the given news has the probability 76% to be fake. Regarding to the explanation, we can obtain it from both attribute and statement analysis. Aided by MIMIC, for this example, we know that "Statement" plays the most important role compared with others. Through ATTN, we can easily check those highlighted words, such as "invited" and "Russia", with different darkness, which would also show the contribution scores when mouse is hovering around. PERT gives users a clear view about which linguistic features contribute to fake and which to true. In the example, we observe that features "Propn Ratio", "Adjective Ratio" and "Noun Ratio" mainly contribute this news to be fake.

User Interface further shows supporting examples and visualized trees for users to better understand the system. As shown in Figure 6, we give two supporting news for instance, where one is retrieved based on the important attributes (Context & Statement) from MIMIC and the other is obtained by matching significant word ("Obama") from ATTN. For the support extraction with MIMIC, we also attach a similarity score, indicating how much attribute information it overlaps with the input one. Besides, 80 decision trees are visualized with interactive diagrams and highlight the activated path of each tree regarding to the input. In Fig. 5.3, we only show one decision tree for example. We can see that each decision tree can be expanded or compressed flexibly, which allows users to track the decision process closely. Given a certain news, each decision tree has only one activated path, corresponding to one specific decision attached at the end of the path with relevant contribution score. Those visualized trees largely enhance the model explainability.

5.3 Server/Back-end

This development is done using FLASK which is a python based web-framework. We use JSON payloads for API calls between client and server. Client request is first validated to check input is in the required format. Then typical preprocessing is done on the input to remove non-ASCII characters, converting to lower case etc. Calls are made to individual models (*MIMIC*, *ATTN* and *PERT*) to get their prediction and explanations. Based on model outputs, additional post processing is done to normalize the outputs and validate word, position info given by *ATT* model for attention scores of n-grams importance. Finaly, all these outputs are sent to the client.

5.4 Effectiveness

To demonstrate the effectiveness of XFAKE [147] in real-world or in other words to understand how different amounts of explanation provided by XFAKE would affect user performance and understanding in assessing the veracity of news statements, we conduct relevant human evaluations by Amazon Mechanical Turk (AMT), with 147 valid testing users in total covering diversified gender, age and education level. Specifically, this experiment was designed to address the following primary research questions:

- How does the amount of explanation information affect human performance in assessing the veracity of news statements?
- How does the amount of explanation information affect user understanding of our models?

Thus, we designed an experiment to test different types and levels of explanation detail in order to evaluate the effects on participant performance and model understanding. The involved user tasks include *Fact Check* and *Prediction Guess*, where the first one is to test the usefulness of the generated explanation and the second one is to indicate the users understanding towards the system. The evaluation metrics are accuracy and time for user prediction.

Overall, the human study results showed a clear trade-off between the speed and accuracy regarding to generated explanations. On one hand, explanation does help users better understand and predict system behavior. On the other hand, explanation would take users more time to review and interpret detection results for benefits.

5.5 Limitation

Current system is limited to the language based (text) inputs and can be extended to vision to make it support multi-model news data which could be a promising extension of the system. Also, current system is limited to political news data as our models are trained on dataset crawled from Politifact website [5]. Thus, extending this system to support multi-source data (i.e not limited to political news), is already the work under progress where we developed additional explanation methodologies for the models trained on new data crawled from Snopes [1] which has news from multiple domains like - Religion, Education, Sports etc.

6. DISCUSSION AND FUTURE WORK

The majority of this thesis has focused on developing separate models for explainable detection of language and vision based fake news that can enhance user trust on our system. Also another major innovation in this thesis was showing the use of explanations in model refinement process by taking advantage of an active learning approach which actually made this refinement feasible. I would like to conclude this thesis with the future direction. My future work involves human-in-the-loop approaches for improving the model interpretations dynamically for multi-modal data. My research will progress along the following paths.

6.1 Multi-Source Fake News Detection

Currently for language based fake news, we only use single source from Politifact.com for detection. To further enhance the detection accuracy and reasonable interpretations, we would like to do the fact-checking from multiple news sources so I plan to incorporate multiple sources for fake news detection. This would make detection results be more solid, and let interpretations become more convincing. I have begun work in this direction by creating a dataset that contains relavent articles (crawled from Google search results) for each news claim of Snopes dataset. I will further model the explainable detection methods such that they also use article source, article content information in addition to the claim content. Thus we can explain the prediction from multiple perspectives like importance of article source, article content, claim content, claim source.

6.2 Multi-Modal Fake News Detection

At present, the news data we considered only involves either text or images. However, we can find many news on web with both image and text. Hence, I plan to incorporate multiple modalities for future detection system. However, due to lack of such datasets, I will creating a dataset using Twitter API and later focus on developing explainable deep learning methods that take cues from both image and corresponding text to make a decision.

6.3 Active-Source Fake News Detection

All our current detection methods focus exclusively on the algorithms and not on human-computer interaction part. This can have impact on the machine performance when tackling real world data. So I would like to explore human-in-the-loop machine learning practices where humans act as active source to optimize the entire learning process, including techniques for annotation. As it's costly to get human input on every data point, and so we need strategies for deciding which data points are the most important for human review. I have begun work in this direction by proposing an active learning criteria (discussed in chapter 3) to select important candidates that require human annotation. Although that work is limited to vision based fake news but i would like to use similar approaches for language as well. Also, i would like to enable human involvement with the right interfaces as shown in Fig. 6.1 which can expedite the efficient labeling of tricky or novel data that a machine can't process, reducing the potential for data-related errors [148].

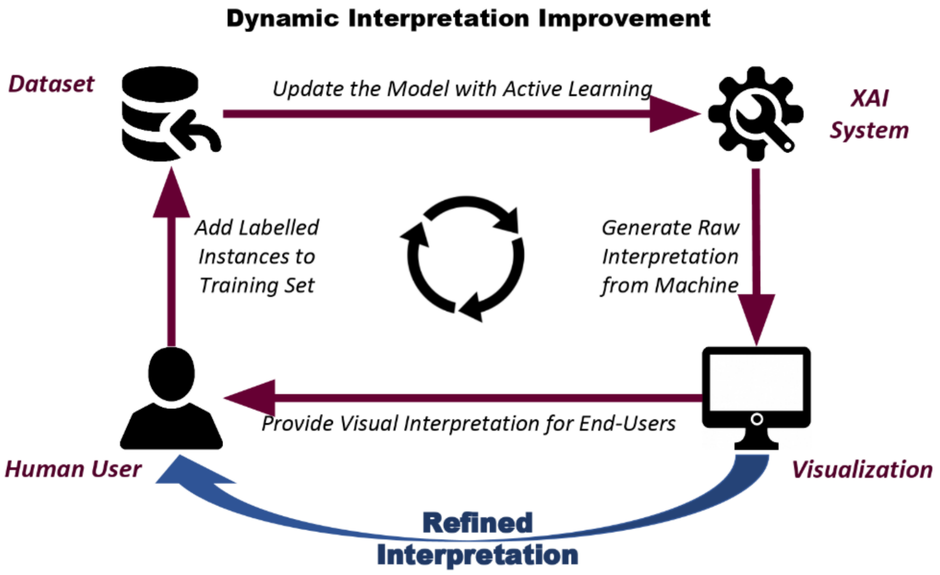

Dynamic Interpretation Improvement
Dataset
Update the Model with Active Learning
XAI System
Add Labelled Instances to Training Set
Generate Raw Interpretation from Machine
Provide Visual Interpretation for End-Users
Human User
Visualization
Refined Interpretation

Figure 6.1: Human-in-the-loop pipeline for refining model interpretations dynamically.

REFERENCES

[1] https://www.snopes.com/. Accessed: 2019-03-30.

[2] K. Shu, H. R. Bernard, and H. Liu, "Studying fake news via network analysis: detection and mitigation," in *Emerging Research Challenges and Opportunities in Computational Social Network Analysis and Mining*, pp. 43–65, Springer, 2019.

[3] M. Du, N. Liu, and X. Hu, "Techniques for interpretable machine learning," *arXiv preprint arXiv:1808.00033*, 2018.

[4] L. H. Gilpin, D. Bau, B. Z. Yuan, A. Bajwa, M. Specter, and L. Kagal, "Explaining explanations: An overview of interpretability of machine learning," in *2018 IEEE 5th International Conference on data science and advanced analytics (DSAA)*, pp. 80–89, IEEE, 2018.

[5] http://www.politifact.com/. Accessed: 2018-11-30.

[6] H. Allcott and M. Gentzkow, "Social media and fake news in the 2016 election," *Journal of economic perspectives*, vol. 31, no. 2, pp. 211–36, 2017.

[7] K. Shu, A. Sliva, S. Wang, J. Tang, and H. Liu, "Fake news detection on social media: A data mining perspective," *ACM SIGKDD Explorations Newsletter*, vol. 19, no. 1, pp. 22–36, 2017.

[8] https://markets.businessinsider.com/news/stocks/tesla-stock-price-is-falling-after-elon-musk-jokes-about-the-company-going-bankrupt-2018-4-1020247710?utm_source=intlutm_medium =ingest. Accessed: 2019-09-25.

[9] https://www.cnbc.com/2018/10/05/tesla-shares-drop-nearly-5percent-after-musk-mocks-sec-on-twitter.html. Accessed: 2019-09-25.

[10] https://tinyurl.com/y8dckwhr. Accessed: 2019-09-25.

[11] DeepFake, "https://github.com/iperov/deepfacelab," 2019.

[12] https://www.buzzfeed.com/. Accessed: 2019-09-25.

[13] https://www.theverge.com/tldr/2018/4/17/17247334/ai-fake-news-video-barack-obama-jordan-peele-buzzfeed. Accessed: 2019-09-25.

[14] https://www.snopes.com/fact-check/trump-second-coming-king-israel/. Accessed: 2019-09-25.

[15] https://www.snopes.com/fact-check/snapchat-evidence-photos/. Accessed: 2019-09-25.

[16] https://www.msnbc.com/hallie-jackson/watch/fake-obama-warning-about-deep-fakes-goes-viral-1214598723984. Accessed: 2019-09-25.

[17] C. R. Sunstein, *On rumors: How falsehoods spread, why we believe them, and what can be done*. Princeton University Press, 2014.

[18] D. M. Lazer, M. A. Baum, Y. Benkler, A. J. Berinsky, K. M. Greenhill, F. Menczer, M. J. Metzger, B. Nyhan, G. Pennycook, D. Rothschild, *et al.*, “The science of fake news,” *Science*, vol. 359, no. 6380, pp. 1094–1096, 2018.

[19] M. Fernandez and H. Alani, “Online misinformation: Challenges and future directions,” in *Companion Proceedings of the The Web Conference 2018*, pp. 595–602, International World Wide Web Conferences Steering Committee, 2018.

[20] P. Hernon, “Disinformation and misinformation through the internet: Findings of an exploratory study,” *Government Information Quarterly*, vol. 12, no. 2, pp. 133–139, 1995.

[21] S. Volkova, K. Shaffer, J. Y. Jang, and N. Hodas, “Separating facts from fiction: Linguistic models to classify suspicious and trusted news posts on twitter,” in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 647–653, 2017.

[22] S. Kumar, R. West, and J. Leskovec, “Disinformation on the web: Impact, characteristics, and detection of wikipedia hoaxes,” in *Proceedings of the 25th international conference on World Wide Web*, pp. 591–602, International World Wide Web Conferences Steering Committee, 2016.

[23] V. L. Rubin, Y. Chen, and N. J. Conroy, "Deception detection for news: three types of fakes," in *Proceedings of the 78th ASIS&T Annual Meeting: Information Science with Impact: Research in and for the Community*, p. 83, American Society for Information Science, 2015.

[24] N. J. Conroy, V. L. Rubin, and Y. Chen, "Automatic deception detection: Methods for finding fake news," *Proceedings of the Association for Information Science and Technology*, vol. 52, no. 1, pp. 1–4, 2015.

[25] V. Rubin, N. Conroy, Y. Chen, and S. Cornwell, "Fake news or truth? using satirical cues to detect potentially misleading news," in *Proceedings of the second workshop on computational approaches to deception detection*, pp. 7–17, 2016.

[26] S. Volkova and J. Y. Jang, "Misleading or falsification: Inferring deceptive strategies and types in online news and social media," in *Companion Proceedings of the The Web Conference 2018*, pp. 575–583, International World Wide Web Conferences Steering Committee, 2018.

[27] S. C. Woolley and P. N. Howard, *Computational propaganda: political parties, politicians, and political manipulation on social media*. Oxford University Press, 2018.

[28] https://tinyurl.com/y9kegobd. Accessed: 2019-09-25.

[29] https://tinyurl.com/ybs4tgpg. Accessed: 2019-09-25.

[30] S. Vosoughi, D. Roy, and S. Aral, "The spread of true and false news online," *Science*, vol. 359, no. 6380, pp. 1146–1151, 2018.

[31] P. N. Howard and B. Kollanyi, "Bots,# strongerin, and# brexit: computational propaganda during the uk-eu referendum," *Available at SSRN 2798311*, 2016.

[32] E. Ferrara, "Disinformation and social bot operations in the run up to the 2017 french presidential election," *First Monday*, vol. 22, no. 8, 2017.

[33] K. Rapoza, "Can 'fake news' impact the stock market?," *Pridobljeno iz www. forbes. com/sites/kenrapoza/2017/02/26/can-fake-news-impact-the-stock-market/(9. 7. 2018)*, 2017.

[34] https://tinyurl.com/z38z5zh. Accessed: 2019-09-25.

[35] E. Ferrara, O. Varol, C. Davis, F. Menczer, and A. Flammini, "The rise of social bots," *Communications of the ACM*, vol. 59, no. 7, pp. 96–104, 2016.

[36] C. Shao, G. L. Ciampaglia, O. Varol, K.-C. Yang, A. Flammini, and F. Menczer, "The spread of low-credibility content by social bots," *Nature communications*, vol. 9, no. 1, p. 4787, 2018.

[37] C. Shao, P.-M. Hui, L. Wang, X. Jiang, A. Flammini, F. Menczer, and G. L. Ciampaglia, "Anatomy of an online misinformation network," *PloS one*, vol. 13, no. 4, p. e0196087, 2018.

[38] F. Cardoso Durier da Silva, R. Vieira, and A. C. Garcia, "Can machines learn to detect fake news? a survey focused on social media," in *Proceedings of the 52nd Hawaii International Conference on System Sciences*, 2019.

[39] E. C. Tandoc Jr, Z. W. Lim, and R. Ling, "Defining "fake news" a typology of scholarly definitions," *Digital journalism*, vol. 6, no. 2, pp. 137–153, 2018.

[40] I. Y. R. Pratiwi, R. A. Asmara, and F. Rahutomo, "Study of hoax news detection using naïve bayes classifier in indonesian language," in *2017 11th International Conference on Information & Communication Technology and System (ICTS)*, pp. 73–78, IEEE, 2017.

[41] C. Castillo, M. Mendoza, and B. Poblete, "Information credibility on twitter," in *Proceedings of the 20th international conference on World wide web*, pp. 675–684, ACM, 2011.

[42] K. Shu, X. Zhou, S. Wang, R. Zafarani, and H. Liu, "The role of user profile for fake news detection," *arXiv preprint arXiv:1904.13355*, 2019.

[43] H. Guo, J. Cao, Y. Zhang, J. Guo, and J. Li, “Rumor detection with hierarchical social attention network,” in *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pp. 943–951, ACM, 2018.

[44] Z. Jin, J. Cao, Y. Zhang, and J. Luo, “News verification by exploiting conflicting social viewpoints in microblogs,” in *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[45] F. Monti, F. Frasca, D. Eynard, D. Mannion, and M. M. Bronstein, “Fake news detection on social media using geometric deep learning,” *arXiv preprint arXiv:1902.06673*, 2019.

[46] K. Shu, D. Mahudeswaran, S. Wang, and H. Liu, “Hierarchical propagation networks for fake news detection: Investigation and exploitation,” *arXiv preprint arXiv:1903.09196*, 2019.

[47] K. Shu, S. Wang, and H. Liu, “Beyond news contents: The role of social context for fake news detection,” in *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, pp. 312–320, ACM, 2019.

[48] H. H. Nguyen, J. Yamagishi, and I. Echizen, “Capsule-forensics: Using capsule networks to detect forged images and videos,” in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019.

[49] H. H. Nguyen, T. Tieu, H.-Q. Nguyen-Son, V. Nozick, J. Yamagishi, and I. Echizen, “Modular convolutional neural network for discriminating between computer-generated images and photographic images,” in *Proceedings of the 13th International Conference on Availability, Reliability and Security*, 2018.

[50] D. Afchar, V. Nozick, J. Yamagishi, and I. Echizen, “Mesonet: a compact facial video forgery detection network,” in *2018 IEEE International Workshop on Information Forensics and Security (WIFS)*, 2018.

[51] Y. Li, M.-C. Chang, H. Farid, and S. Lyu, “In ictu oculi: Exposing ai generated fake face videos by detecting eye blinking,” *IEEE Workshop on Information Forensics and Security (WIFS)*, 2018.

[52] Y. Li and S. Lyu, "Exposing deepfake videos by detecting face warping artifacts," *Workshop on Media Forensics (in conjuction with CVPR)*, 2019.

[53] M. Huh, A. Liu, A. Owens, and A. A. Efros, "Fighting fake news: Image splice detection via learned self-consistency," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.

[54] H. Li, B. Li, S. Tan, and J. Huang, "Detection of deep network generated images using disparities in color components," *arXiv preprint arXiv:1808.07276*, 2018.

[55] D. Cozzolino, J. Thies, A. Rössler, C. Riess, M. Nießner, and L. Verdoliva, "Forensictransfer: Weakly-supervised domain adaptation for forgery detection," *arXiv preprint arXiv:1812.02510*, 2018.

[56] Y. Xia, X. Cao, F. Wen, G. Hua, and J. Sun, "Learning discriminative reconstructions for unsupervised outlier removal," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2015.

[57] X. Zhou and R. Zafarani, "Fake news: A survey of research, detection methods, and opportunities," *arXiv preprint arXiv:1812.00315*, 2018.

[58] X. Zhou, R. Zafarani, K. Shu, and H. Liu, "Fake news: Fundamental theories, detection strategies and challenges," in *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, pp. 836–837, ACM, 2019.

[59] M. Potthast, J. Kiesel, K. Reinartz, J. Bevendorff, and B. Stein, "A stylometric inquiry into hyperpartisan and fake news," *arXiv preprint arXiv:1702.05638*, 2017.

[60] C. Guo, J. Cao, X. Zhang, K. Shu, and M. Yu, "Exploiting emotions for fake news detection on social media," *arXiv preprint arXiv:1903.01728*, 2019.

[61] J. Fairbanks, N. Fitch, N. Knauf, and E. Briscoe, "Credibility assessment in the news: Do we need to read," in *Proc. of the MIS2 Workshop held in conjuction with 11th Int'l Conf. on Web Search and Data Mining*, pp. 799–800, 2018.

[62] B. D. Horne and S. Adali, “This just in: Fake news packs a lot in title, uses simpler, repetitive content in text body, more similar to satire than real news,” in *Eleventh International AAAI Conference on Web and Social Media*, 2017.

[63] J. W. Pennebaker, R. L. Boyd, K. Jordan, and K. Blackburn, “The development and psychometric properties of liwc2015,” tech. rep., 2015.

[64] K. Popat, S. Mukherjee, A. Yates, and G. Weikum, “Declare: Debunking fake news and false claims using evidence-aware deep learning,” *arXiv preprint arXiv:1809.06416*, 2018.

[65] H. Karimi, P. Roy, S. Saba-Sadiya, and J. Tang, “Multi-source multi-class fake news detection,” in *Proceedings of the 27th International Conference on Computational Linguistics*, pp. 1546–1557, 2018.

[66] H. Karimi and J. Tang, “Learning hierarchical discourse-level structure for fake news detection,” *arXiv preprint arXiv:1903.07389*, 2019.

[67] S. Baird, D. Sibley, and Y. Pan, “Talos targets disinformation with fake news challenge victory,” *Fake News Challenge*, 2017.

[68] A. Hanselowski, P. Avinesh, B. Schiller, and F. Caspelherr, “Description of the system developed by team athene in the fnc-1,” tech. rep., Technical Report. Technical report, 2017.

[69] W. Y. Wang, “" liar, liar pants on fire": A new benchmark dataset for fake news detection,” *arXiv preprint arXiv:1705.00648*, 2017.

[70] S. Hosseinimotlagh and E. E. Papalexakis, “Unsupervised content-based identification of fake news articles with tensor decomposition ensembles,” *MIS2, Marina Del Rey, CA, USA*, 2018.

[71] E. Tacchini, G. Ballarin, M. L. Della Vedova, S. Moret, and L. de Alfaro, “Some like it hoax: Automated fake news detection in social networks,” *arXiv preprint arXiv:1704.07506*, 2017.

[72] L. Wu and H. Liu, "Tracing fake-news footprints: Characterizing social media messages by how they propagate," in *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, pp. 637–645, ACM, 2018.

[73] Y. Liu and Y.-F. B. Wu, "Early detection of fake news on social media through propagation path classification with recurrent and convolutional networks," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[74] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv preprint arXiv:1412.3555*, 2014.

[75] S. Yang, K. Shu, S. Wang, R. Gu, F. Wu, and H. Liu, "Unsupervised fake news detection on social media: A generative approach," in *Proceedings of 33rd AAAI Conference on Artificial Intelligence*, 2019.

[76] C. Robert and G. Casella, *Monte Carlo statistical methods*. Springer Science & Business Media, 2013.

[77] N. Ruchansky, S. Seo, and Y. Liu, "Csi: A hybrid deep model for fake news detection," in *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pp. 797–806, ACM, 2017.

[78] A. Nguyen, J. Yosinski, and J. Clune, "Deep neural networks are easily fooled: High confidence predictions for unrecognizable images," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 427–436, 2015.

[79] N. Liu, H. Yang, and X. Hu, "Adversarial detection with model interpretation," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 1803–1811, ACM, 2018.

[80] K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep inside convolutional networks: Visualising image classification models and saliency maps," *arXiv preprint arXiv:1312.6034*, 2013.

[81] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller, “Striving for simplicity: The all convolutional net,” *arXiv preprint arXiv:1412.6806*, 2014.

[82] S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. Müller, and W. Samek, “On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation,” *PloS one*, vol. 10, no. 7, p. e0130140, 2015.

[83] M. Ancona, E. Ceolini, C. Öztireli, and M. Gross, “Towards better understanding of gradient-based attribution methods for deep neural networks,” *arXiv preprint arXiv:1711.06104*, 2017.

[84] P. Dabkowski and Y. Gal, “Real time image saliency for black box classifiers,” in *Advances in Neural Information Processing Systems*, pp. 6967–6976, 2017.

[85] R. C. Fong and A. Vedaldi, “Interpretable explanations of black boxes by meaningful perturbation,” in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3429–3437, 2017.

[86] M. Du, N. Liu, Q. Song, and X. Hu, “Towards explanation of dnn-based prediction with guided feature inversion,” in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 1358–1367, ACM, 2018.

[87] M. Du, N. Liu, F. Yang, S. Ji, and X. Hu, “On attribution of recurrent neural network predictions via additive decomposition,” in *The World Wide Web Conference*, pp. 383–393, ACM, 2019.

[88] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, “Learning deep features for discriminative localization,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2921–2929, 2016.

[89] R. R. Selvaraju, A. Das, R. Vedantam, M. Cogswell, D. Parikh, and D. Batra, “Grad-cam: Why did you say that?,” *arXiv preprint arXiv:1611.07450*, 2016.

[90] C. Molnar *et al.*, "Interpretable machine learning: A guide for making black box models explainable," *E-book at< https://christophm. github. io/interpretable-ml-book/>, version dated*, vol. 10, 2018.

[91] D. Alvarez-Melis and T. S. Jaakkola, "A causal framework for explaining the predictions of black-box sequence-to-sequence models," *arXiv preprint arXiv:1707.01943*, 2017.

[92] L. Arras, G. Montavon, K.-R. Müller, and W. Samek, "Explaining recurrent neural network predictions in sentiment analysis," *arXiv preprint arXiv:1706.07206*, 2017.

[93] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[94] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio, "Show, attend and tell: Neural image caption generation with visual attention," in *International conference on machine learning*, pp. 2048–2057, 2015.

[95] D. A. Hudson and C. D. Manning, "Compositional attention networks for machine reasoning," *arXiv preprint arXiv:1803.03067*, 2018.

[96] M. Denil, A. Demiraj, and N. De Freitas, "Extraction of salient sentences from labelled documents," *arXiv preprint arXiv:1412.6815*, 2014.

[97] Y. Hechtlinger, "Interpretation of prediction models using the input gradient," *arXiv preprint arXiv:1611.07634*, 2016.

[98] W. J. Murdoch, P. J. Liu, and B. Yu, "Beyond word importance: Contextual decomposition to extract interactions from lstms," *arXiv preprint arXiv:1801.05453*, 2018.

[99] W. J. Murdoch and A. Szlam, "Automatic rule extraction from long short term memory networks," *arXiv preprint arXiv:1702.02540*, 2017.

[100] A. Kádár, G. Chrupała, and A. Alishahi, "Representation of linguistic form and function in recurrent neural networks," *Computational Linguistics*, vol. 43, no. 4, pp. 761–780, 2017.

[101] J. Li, W. Monroe, and D. Jurafsky, "Understanding neural networks through representation erasure," *arXiv preprint arXiv:1612.08220*, 2016.

[102] N. Poerner, H. Schütze, and B. Roth, "Evaluating neural network explanation methods using hybrid documents and morphological agreement," *arXiv preprint arXiv:1801.06422*, 2018.

[103] M. T. Ribeiro, S. Singh, and C. Guestrin, "Why should i trust you?: Explaining the predictions of any classifier," in *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 1135–1144, ACM, 2016.

[104] M. T. Ribeiro, S. Singh, and C. Guestrin, "Anchors: High-precision model-agnostic explanations," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[105] Z. Che, S. Purushotham, R. Khemani, and Y. Liu, "Distilling knowledge from deep networks with applications to healthcare domain," *arXiv preprint arXiv:1512.03542*, 2015.

[106] P. Rafi, A. Pakbin, and S. K. Pentyala, "Interpretable deep learning framework for predicting all-cause 30-day icu readmissions," *Texas A&M University*, 2018.

[107] R. Caruana, Y. Lou, J. Gehrke, P. Koch, M. Sturm, and N. Elhadad, "Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission," in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1721–1730, ACM, 2015.

[108] G. Vandewiele, O. Janssens, F. Ongenae, F. De Turck, and S. Van Hoecke, "Genesim: genetic extraction of a single, interpretable model," *arXiv preprint arXiv:1611.05722*, 2016.

[109] O. Bastani, C. Kim, and H. Bastani, "Interpretability via model extraction," *arXiv preprint arXiv:1706.09773*, 2017.

[110] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *International Conference on Learning Representations (ICLR)*, 2015.

[111] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *International Conference on Learning Representations (ICLR)*, 2016.

[112] A. Dosovitskiy and T. Brox, "Generating images with perceptual similarity metrics based on deep networks," in *Advances in neural information processing systems (NIPS)*, 2016.

[113] A. Rössler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Nießner, "Faceforensics++: Learning to detect manipulated facial images," *arXiv preprint arXiv:1901.08971*, 2019.

[114] J. Thies, M. Zollhofer, M. Stamminger, C. Theobalt, and M. Nießner, "Face2face: Real-time face capture and reenactment of rgb videos," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2387–2395, 2016.

[115] Faceswap, "https://github.com/marekkowalski/faceswap/," 2019.

[116] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep learning face attributes in the wild," in *Proceedings of the IEEE international conference on computer vision (ICCV)*, 2015.

[117] Y. Choi, M. Choi, M. Kim, J.-W. Ha, S. Kim, and J. Choo, "Stargan: Unified generative adversarial networks for multi-domain image-to-image translation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[118] D. P. Kingma and P. Dhariwal, "Glow: Generative flow with invertible 1x1 convolutions," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

[119] S. Iizuka, E. Simo-Serra, and H. Ishikawa, "Globally and locally consistent image completion," *ACM Transactions on Graphics (ToG)*, 2017.

[120] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, and T. S. Huang, "Generative image inpainting with contextual attention," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[121] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, *et al.*, "Imagenet large scale visual recognition challenge," *International journal of computer vision*, vol. 115, no. 3, pp. 211–252, 2015.

[122] B. Bayar and M. C. Stamm, "A deep learning approach to universal image manipulation detection using a new convolutional layer," in *Proceedings of the 4th ACM Workshop on Information Hiding and Multimedia Security*, 2016.

[123] D. Cozzolino, G. Poggi, and L. Verdoliva, "Recasting residual-based local descriptors as convolutional neural networks: an application to image forgery detection," in *Proceedings of the 5th ACM Workshop on Information Hiding and Multimedia Security*, 2017.

[124] N. Rahmouni, V. Nozick, J. Yamagishi, and I. Echizen, "Distinguishing computer graphics from natural images using convolution neural networks," in *2017 IEEE Workshop on Information Forensics and Security (WIFS)*, 2017.

[125] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, 2017.

[126] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*, Springer, 2015.

[127] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[128] M. Du, N. Liu, F. Yang, and X. Hu, "Learning credible deep neural networks with rationale regularization," in *IEEE International Conference on Data Mining (ICDM)*, 2019.

[129] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017.

[130] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, pp. 5998–6008, 2017.

[131] M. Ott, Y. Choi, C. Cardie, and J. T. Hancock, "Finding deceptive opinion spam by any stretch of the imagination," in *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies-volume 1*, pp. 309–319, Association for Computational Linguistics, 2011.

[132] V. Pérez-Rosas and R. Mihalcea, "Experiments in open domain deception detection," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 1120–1125, 2015.

[133] A. Vlachos and S. Riedel, "Fact checking: Task definition and dataset construction," in *Proceedings of the ACL 2014 Workshop on Language Technologies and Computational Social Science*, pp. 18–22, 2014.

[134] W. Ferreira and A. Vlachos, "Emergent: a novel data-set for stance classification," in *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: Human language technologies*, pp. 1163–1168, 2016.

[135] http://blogs.channel4.com/factcheck/. Accessed: 2018-01-30.

[136] K. Crammer and Y. Singer, "On the algorithmic implementation of multiclass kernel-based vector machines," *Journal of machine learning research*, vol. 2, no. Dec, pp. 265–292, 2001.

[137] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[138] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional lstm and other neural network architectures," *Neural networks*, vol. 18, no. 5-6, pp. 602–610, 2005.

[139] A. Moschitti, B. Pang, and W. Daelemans, "Proceedings of the 2014 conference on empirical methods in natural language processing (emnlp),"

[140] T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Efficient estimation of word representations in vector space,” *arXiv preprint arXiv:1301.3781*, 2013.

[141] J. Pennington, R. Socher, and C. Manning, “Glove: Global vectors for word representation,” in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pp. 1532–1543, 2014.

[142] “Explaining tree ensembles with eli5.” https://eli5.readthedocs.io/en/0.2/autodocs/sklearn.html. Accessed: 2019-01-30.

[143] G. Louppe, L. Wehenkel, A. Sutera, and P. Geurts, “Understanding variable importances in forests of randomized trees,” in *Advances in neural information processing systems*, pp. 431–439, 2013.

[144] P. Hall, N. Gill, and M. Chan, “Practical techniques for interpreting machine learning models: Introductory open source examples using python, h2o, and xgboost,” 2018.

[145] B. Kim, C. Rudin, and J. A. Shah, “The bayesian case model: A generative approach for case-based reasoning and prototype classification,” in *Advances in Neural Information Processing Systems*, pp. 1952–1960, 2014.

[146] B. Kim, R. Khanna, and O. O. Koyejo, “Examples are not enough, learn to criticize! criticism for interpretability,” in *Advances in Neural Information Processing Systems*, pp. 2280–2288, 2016.

[147] F. Yang, S. K. Pentyala, S. Mohseni, M. Du, H. Yuan, R. Linder, E. D. Ragan, S. Ji, and X. B. Hu, “Xfake: Explainable fake news detector with visualizations,” in *The World Wide Web Conference*, pp. 3600–3604, ACM, 2019.

[148] R. Munro, “Human-in-the-loop machine learning.”