Detecting Deepfakes using both ANN and CNN

Vishal Kodam
Computer and Information Sciences
Suny Polytechnic Institute
Utica, New York, USA
kodamv@sunypoly.edu

Mamatha Narri
Computer and Information Sciences
Suny Polytechnic Institute
Utica, New York, USA
narrim@sunypoly.edu

Bavitha Raavi
Computer and Information Sciences
Suny Polytechnic Institute
Utica, New York, USA
raavib@sunypoly.edu

Anil Reddy Chepyala
Computer and Information Sciences
Suny Polytechnic Institute
Utica, New York, USA
chepyaa@sunypoly.edu

***Abstract*—Our lives are now easier and more productive thanks to enormous breakthroughs made by technology. The technology behind many of these developments, machine learning, has been essential in this development. But the popularity of DeepFakes, which are manufactured and fake videos and images, has brought attention to the potential abuse of machine learning technology for bad intentions. This article investigates how DeepFakes, a subset of machine learning that are produced using deep learning technology, are produced and how they are detected. While deep learning has aided in the resolution of challenging issues in a variety of fields, it has also produced DeepFakes, which pose a risk to national security, privacy, and democracy. The main goal of this study is to increase awareness of the risks associated with fake online information among young people who have grown up with digital influences. The article outlines a DeepFake video creation algorithm and suggests several ways to spot it. The paper also covers the advantages of DeepFake development and detection and how they may be used effectively without harm. Here we will be implementing ResNext for feature extraction and LSTM algorithm for sequential analysis and detection.**

Keywords—DeepFakes, Convolutional Neural Network, Deep Learning, LSTM, CNN, ANN

I. INTRODUCTION

The making and distribution of digital videos is now easier than ever because to the improving smartphone cameras, the widespread availability of fast internet connections, and the ever-expanding reach of social media. Deep learning has become so powerful because to the increase in processing power that it was previously regarded to be unthinkable. This has brought about new difficulties, as with any transformative technology. Deep generative adversarial models with the ability to modify audio and video samples are what is known as "DeepFake" content. It has been quite usual to spread the DF through social media platforms, which encourages spamming and the spread of false information. These kinds of DF will intimidate and mislead the general public.

To solve this problem, it is essential to identify deepfakes. In order to effectively distinguish computer-generated phony videos from real videos, we introduce a revolutionary deep learning-based method. To stop deepfakes from spreading online, it is essential to develop technologies capable of identifying fake videos.

It is essential to understand how the Generative Adversarial Network (GAN) generates deepfakes (DF) in order to detect them. The procedure involves entering a video and a picture of a certain person (referred to as the "target"), which results in the creation of a second film with the target's face replaced with that of another person (referred to as the "source"). Deep adversarial neural networks, the cornerstone of DF, are trained on target movies and face photos to automatically map source faces and facial emotions to the target. The resulting videos can achieve a high level of realism with the right post-processing. In order to reconstruct the video, GAN splits the video into frames and replaces the input image in each frame, typically using autoencoders.

We describe a brand-new deep learning-based method for accurately separating DF videos from actual ones. Our approach is based on the characteristics of DF videos and uses the same procedure as DF by GAN. The DF technique can only synthesize face pictures of a certain size because to computational constraints and production time, and these images must go through affine warping in order to match the original face's configuration. The twisted face area and surrounding context have different resolutions, which causes identifiable artifacts in the deepfake video output.

By comparing the created face areas and the areas around them, our approach finds such artifacts. Using a ResNext Convolutional Neural Network (CNN), we break the video up into frames and extract the features. We then utilize a Recurrent Neural Network (RNN) with Long Short Term Memory (LSTM) to capture the temporal discrepancies between frames that GAN introduced during the DF reconstruction. By explicitly modeling the resolution discrepancy in affine face wrappings, we streamline the training of the ResNext CNN model.

Contribution of individuals are as follows:

Anil Chepyala was responsible for data preprocessing and analysis, which involved gathering and preprocessing the dataset used for training and testing the deep fake detection model. He also contributed to the experimentation phase, by fine-tuning the hyperparameters of the model and testing its performance. In addition to this I have also worked on creating the abstract, Introduction and Motivation of the research paper.

Mamatha Narri was responsible for developing the ResNeXt architecture for the deep fake detection model, using PyTorch libraries. She worked on improving the model's accuracy and robustness by fine-tuning the architecture and exploring different variations of the ResNeXt algorithm. Moreover, have also analyzed the model architecture, Hyperparameter tuning and contributed to the Approach of the report and was involved in technical steps to be followed.

Bavitha Raavi was responsible for the evaluation of the deep fake detection model using suitable metrics, integrated ResNeXt and CNN models, tested and resolved issues, and proposed future work to extend the project, including verifying accurate video detection on a sample dataset. She have also contributed to the background and Limitations in the project report.

Vishal Kodam was responsible of the core idea and designed the model interpretability framework for the deep fake detection model. He contributed to the experimental evaluation of the project report which include visualisation, confusion matrix, training loss, etc. and gathered the necessary information from the literature survey of previous research papers.

Throughout the project, each team member collaborated and provided valuable feedback and suggestions to improve the model's accuracy and performance. We also shared responsibilities and helped each other as needed, which led to the successful completion of the project. Overall, this project was a collaborative effort, and the contributions of each team member were crucial in achieving the project objectives.

II. MOTIVATION

Democracy, justice, and public trust are seriously threatened by the deep fake video industry's tremendous expansion and criminal usage. As a result the need for bogus video analysis, detection, and action has increased.

The development of social media and mobile camera technologies has simplified the creation and distribution of digital videos. Deep learning innovations have given rise to previously inconceivable technologies like generative models that can produce lifelike voice, music, images, and videos. These models have been used in a variety of disciplines, such as producing data for medical imaging training and developing text-to-speech tools to make the world more accessible.

As with any revolutionary technology, the growing accessibility of complex deep generative models has brought about new difficulties. These algorithms can create "deep fakes"—manipulated audio and video snippets that are difficult to tell apart from the real thing. Even while some of these fakes could have been produced for amusement, there is growing fear that they could be used to spread false information and hurt people and society. Recent years have seen an increase in the number of fake films that can be easily shared on social media platforms, which has resulted in spamming and the transmission of misleading information. This increase is due to the availability of open-source deep fake generating tools and the desire for domain expertise.These deep fakes have grave repercussions that can range from a fake video of a government figure declaring war to a false film of a well-known celebrity behaving violently. These elaborate fakes have the ability to deceive and frighten regular people.

It has become essential to identify deep fakes in order to address this issue. In order to accurately identify between deep fake movies and real ones, we introduce a revolutionary deep learning-based technique. To detect and stop the spread of deep fakes on the internet, it is crucial that such technology be developed.

III. BACKGROUND

In brief for preprocessing we used **glob** to import videos and the function cv2.VideoCapture is used for reading the videos and getting mean of frames. The other tools include Python Libraries such as NumPy, Pandas, Sklearn for visualisation.

The model combines CNN and RNN to classify videos as deepfake or pristine. Pre-trained ResNext CNN model is used to extract frame-level features. A 1-layer LSTM network with 2048 latent dimensions and 0.4 dropout is trained on these features to analyze the video temporally. ResNext50_32x4d is used for feature extraction. The 2048-dimensional feature vectors after the last pooling layers of ResNext are used as sequential LSTM input. The network is fine-tuned by adding required layers and selecting a proper learning rate. Data Loader is used to load and fit video labels for training.

IV. APPROACH

Deepfakes can be produced using a wide range of techniques, but their detection is difficult. Our detection strategy will play a significant role in limiting the online dissemination of deepfakes. A web-based platform will be made so that users can upload films to be categorized as bogus or authentic. With the potential to be linked into well-known programs like WhatsApp and Facebook, this project might be expanded to include a browser plugin for automatic detection. Our main goal is to identify different deepfakes, such as replacement, retrenchment, and interpersonal ones. Security, user friendliness, correctness, and dependability will all be evaluated. Figure.1 represents the architecture of the proposed model: -

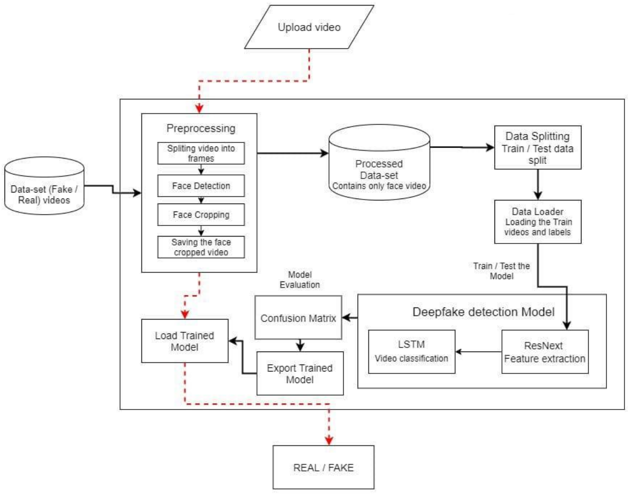
Upload video
Preprocessing
Spliting video into frames
Face Detection
Face Cropping
Saving the face cropped video
Data-set (Fake / Real) videos
Processed Data-set Contains only face video
Data Splitting Train / Test data split
Data Loader Loading the Train videos and labels
Train / Test the Model
Model Evaluation
Confusion Matrix
Deepfake detection Model
LSTM Video classification
ResNext Feature extraction
Load Trained Model
Export Trained Model
REAL / FAKE

Fig1: MODEL ARCHITECTURE

Following are the technical steps involved:

A. *Preprocessing:*

In the preprocessing step of the video, background noise is removed while only the video's facial features are kept. This is accomplished by dividing the video into frames, identifying the face in each frame, and then cropping and merging the frames to create a new video that only contains facial features. During this procedure, the frames without any facial features are disregarded. Based on the average frame count of each video and the GPU's processing capabilities, a threshold value of 150 frames was chosen to assure consistency in the number of frames. For the purpose of illustrating how to utilize LSTM properly, the frames were recorded in a sequential order. The newly created video was saves at 30fps which results in a processed dataset of videos containing only facial features.

B. Model Architecture

Our method combines recurrent neural networks (RNN) and convolutional neural networks (CNN). We extracted the attributes from each video frame using the pre-trained ResNext CNN model, and then we trained an LSTM network to categorize the videos as deepfakes or real. With the aid of a data loader, the labels for the training videos were loaded and fitted into our training model.

ResNext for feature extraction:

For feature extraction in our project, we used ResNext, a pre-trained model enhanced for good performance on deeper neural networks. We specifically used the 50-layer, 32 x 4 dimension resnext50_32x4d model. We added the extra layers that were needed and chose a suitable learning rate to converge the gradient descent of the model in order to fine-tune the network for our needs. The sequential LSTM input used 2048-dimensional feature vectors that were obtained from ResNext's final pooling layers.

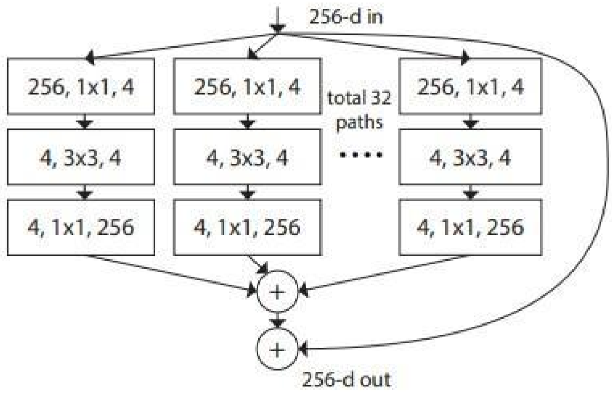

Figure 2. ResNext50 Architecture

LSTM for sequence processing:

A feature vector with 2048 dimensions serves as the LSTM's input. One LSTM layer with 2048 latent dimensions, 2048 hidden layers, and 0.4 dropout probability makes up the model. To examine the temporal structure of the video, the LSTM layer processes frames in a sequential manner. It employs a Leaky ReLU activation function. In order to learn the correlation between the input and output, the model also has a linear layer with 2048 input features and 2 output features. To achieve the desired output size, an adaptive average pooling layer with an output parameter of 1 is used. Using a Sequential Layer, the frames are processed one after the other, with a batch size of four. A SoftMax layer is used to determine the model's confidence during prediction.

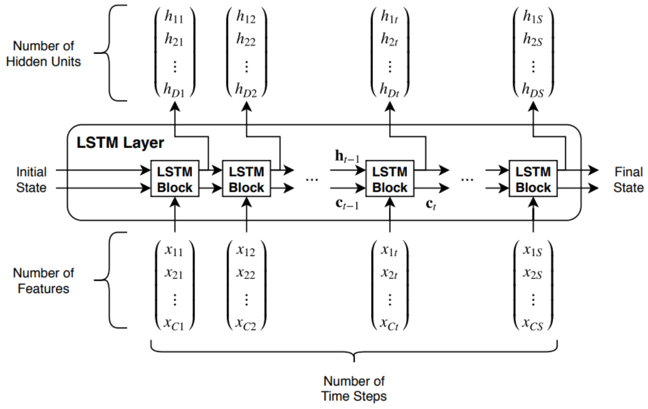
Number of Hidden Units
h_{11} h_{21} ⋮ h_{D1}
h_{12} h_{22} ⋮ h_{D2}
h_{1t} h_{2t} ⋮ h_{Dt}
h_{1S} h_{2S} ⋮ h_{DS}
LSTM Layer
Initial State
LSTM Block
$\mathbf{h}_{t-1}$
$\mathbf{c}_{t-1}$
$\mathbf{c}_t$
Final State
Number of Features
x_{11} x_{21} ⋮ x_{C1}
x_{12} x_{22} ⋮ x_{C2}
x_{1t} x_{2t} ⋮ x_{Ct}
x_{1S} x_{2S} ⋮ x_{CS}
Number of Time Steps

Figure 3. LSTM Architecture

C. Predict:

The new video is initially preprocessed to match the format of the trained model for video prediction. The face is cropped, and the footage is divided into frames. The clipped frames are directly entered into the trained model for detection rather than being stored with the video.

V. EXPERIMENTAL EVALUATION

Our project include the use of Python as programming language and along with few libraries and frameworks. Some of them include OpenCV for image processing techniques and performing data visualisation using Pandas and NumPy. The IDE we used to implement our machine learning models is Google Colab and Jupyter.

For image classification, especially in feature extraction, Pytorch is used. We imported all the videos in a directory into a Python list using the glob function. To maintain uniformity in the number of frames in each video, we used cv2.VideoCapture to read the videos and get the mean number of frames in each video. Based on this mean, we selected 150 frames as the ideal value for creating a new dataset. The videos were then split into frames and each frame was cropped on the face location. The face-cropped frames were then written to a new video using VideoWriter in mp4 format with a resolution of 112 x 112 pixels and 30 frames per second. To make proper use of the LSTM for temporal sequence analysis, we only used the first 150 frames in the new video instead of selecting random frames from the videos. A pretrained ResNext50 CNN Deep Learning model is used for feature extraction and LSTM is used for sequential analysis and prediction using Sklearn library. A confusion matrix is also created to understand the data using Pandas.

There were a lot of research questions before, while and after the project. Following are our questions listed:

- In the beginning of the project it was our challenge of choosing the right dataset.
- How are we going to get a clean dataset containing only of images.

- During research we were confused of which model is to be used for extracting features i.e: ResNet or ResNext.
- Can the model detect a distorted or low quality images accurately?
- How do we solve overfitting model issues?

These question motivated us to conduct more research and experiments on our project, guiding us to the results.

Dataset:

We are using a mixed dataset which consists of equal amount of videos from different dataset sources like FaceForensics++ and Deep fake detection challenge dataset. Our newly prepared dataset contains 50% of the original video and 50% of the manipulated deepfake videos. The dataset is split into 70% train and 30% test set. Using a Python script, the Deep Fake Detection Challenge (DFDC) dataset was preprocessed in this effort to get rid of the audio-altered films. After preprocessing, 3000 Real and 3000 Fake videos were chosen from the DFDC, FaceForensic++, and CelebDF datasets; 1500 of these were from DFDC, 1000 from FaceForensic++, and 500 from CelebDF. A total of 6000 videos made up the dataset. Figure 2 displays the distribution of the datasets.

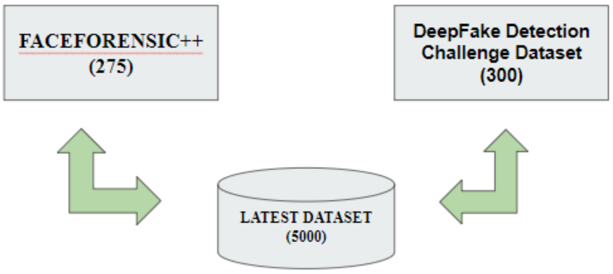

Figure 4. Dataset Collection

Results and Analysis:

Our key agenda was to identify the real time video to be fake or not. Due to loading of less dataset, lack of time and resources we couldn't train and test large portions of data. But with the available dataset and resources we managed to get a good accuracy score in different epochs. Our approach uses ResNext CNN for frame level detection and RNN and LSTM for video classification.
Fig5. shows the confusion matrix which is a bit inaccurate because of taking less dataset.

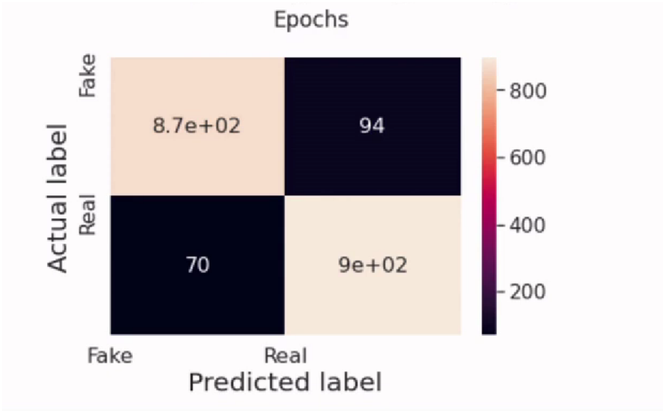

Figure 5. Confusion Matrix

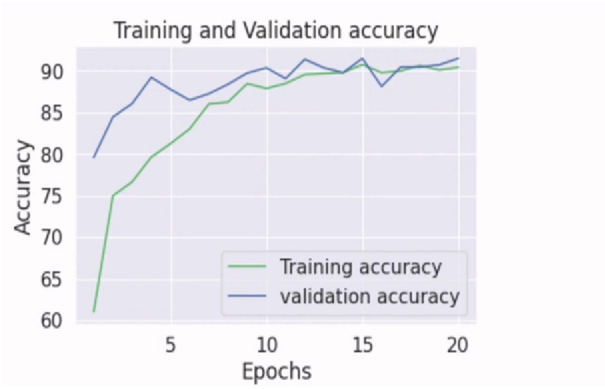

Figure 6. Training accuracy vs. Validation accuracy over the number of epochs

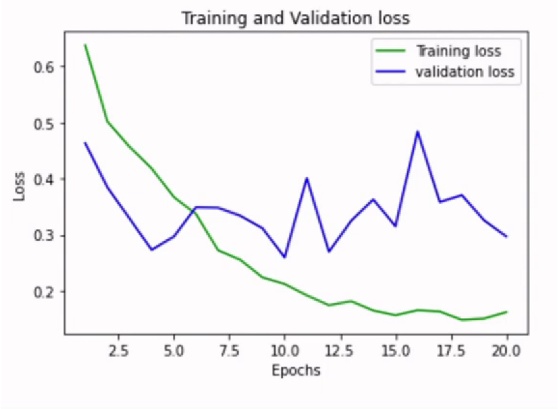

Figure 7. Training loss vs. Validation loss over the number of epochs

TABLE 1

DATASET	NO. OF VIDEOS	SEQUENCE LENGTH	ACCURACY
FaceForensic ++	2000	20	90.95477
Our Dataset	586	20	87.79160
Our Dataset	586	10	84.21461
Our Dataset	586	40	89.34681
Our Dataset	586	60	89.98345

Table1 shows the comparitive accuracy of fake detection by using FaceForensic++ dataset and our preprocessed dataset.

In this paper for DeepFake detection, we have implemented both CNN and ANN model. The Training accuracy vs the validation accuracy is plotted over the number of epochs in Fig.6. and Fig.7. is the plot of Training loss vs. Validation loss over the number of epochs. We have obtained a constant accuracy by changing the sequence length till 40. The final accuracy score obtained in this model is 89.9835 using a limited preprocessed dataset.

VI. LIMITATIONS

Our method cannot identify audio deep fakes since it does not consider audio. To identify audio deep fakes, however, we want to create a future solution.

Due to unavailability of resources and time we couldn't train the model with higher volume of dataset and planned resulting in no ral time detection.

A User Interface cannot be created due to the above stated same reasons failing the vision of our project.

VII. RELATED WORKS

The Media Forensics and DeepFakes paper aims to investigate ways to authenticate visual media, especially deepfakes, due to the emergence of highly realistic fabricated content. The authors analyze current data-driven forensic tools like autoencoders and GANs to highlight their limitations from a forensic analyst's perspective. They also discuss the significant challenges and future research directions in this area.

The Exposing Digital Image Forgeries paper presents a new method for detecting image splicing forgery using GLCM and LBP texture features. The proposed method estimates the illuminant color, detects edges using a canny edge detector, combines GLCM and LBP features, and uses a KNN classifier for classification. The algorithm requires minimal user interaction and can determine whether an image is genuine or forged. The authors evaluate the method on two datasets using the KNN classifier

The survey of deepfake-creating algorithms and suggested strategies to catch them is presented in this research. The authors provide a thorough review of the subject by talking about the difficulties, research trends, and future directions associated with deepfake technology. The creation of fake images and videos that are nearly indistinguishable from real ones has become possible thanks to the development of deepfake algorithms, stressing the need for technology that can automatically detect and evaluate the veracity of digital visual media.

The study presents FPGAN, a GAN-based approach to protect facial privacy. To enhance feature extraction, FPGAN employs a generator with an enhanced U-Net and two discriminators with a seven-layer design. The paper applies FPGAN to the face de-identification of social robots and suggests a novel evaluation process. On various datasets, FPGAN's performance is compared to that of four other methods, and experimental findings demonstrate its superiority to traditional methods.

Exposing DeepFake Videos, In this paper, we describe a new deep learning based method that can effectively distinguish DeepFake videos from the real ones. The paper presents a novel approach utilizing deep learning techniques to accurately identify DeepFake videos from authentic ones. The proposed method employs a specialized Convolutional Neural Network (CNN) model to compare the facial regions of the videos and their surrounding areas, effectively detecting any artifacts present. The researchers evaluated the method on various sets of publicly available DeepFake videos, and the results demonstrate its practical effectiveness.

In MesoNet paper, a compact neural network designed to detect facial video forgery, with a specific focus on Deepfake and Face2Face techniques. Traditional forensic methods are not effective, so MesoNet uses two small networks that concentrate on mesoscopic image properties. The method achieves an accuracy of over 98% for Deepfake and 95% for Face2Face on two datasets.

In Detecting manipulated Facial Images, authors propose a benchmark for facial manipulation detection that includes Deep-Fakes and Face2Face techniques with varying compression levels and sizes. They show that domain-specific knowledge significantly improves forgery detection accuracy, surpassing human observer's performance, even in strong compression. The benchmark is ten times larger than similar datasets, with a hidden test set and millions of manipulated images.

VIII. CONCLUSION

Our study proposes a neural network-based approach for detecting deep fake videos, which is inspired by the way GANs create deep fakes using autoencoders. Our method performs frame-level detection using ResNext CNN and video classification using RNN with LSTM. We anticipate that our approach will achieve high accuracy in real-time data detection.

We tuned the parameters to achieve better efficiency and solve the overfitting issue. Since our dataset was small we couldn't achieve real time detection but given time and resources we can achieve it and also can create a UI.

References

[1] P. Korshunov and S. Marcel, "Deepfakes: a New Threat to Face Recognition? Assessment and Detection," arXiv preprint arXiv:1812.08685, 2018.

[2] D.Citron, "How DeepFake Undermine Truth and Threaten Democracy," 2019. [Online]. Available: https://www.ted.com

[3] R. Cellan-Jones, "Deepfake Videos Double in Nine Months," 2019.[Online].Available:https://www.bbc.com/news/technology-49961089

[4] BBC Bitesize, "Deepfakes: What Are They and Why Would I Make One?" 2019. [Online]. Available: https://www.bbc.co.uk/bitesize/articles/zfkwcqt

[5] A. Swaminathan, M. Wu and K.J.R. Liu, "Digital Image Forensics via Intrinsic Fingerprints," IEEE Transactions on Information Forensics and Security, vol. 3, no. 1, pp. 101–117, 2008.

[6] https://www.kaggle.com/datasets/sorokin/faceforensics

[7] https://www.kaggle.com/c/deepfake-detection-challenge/data

[8] Sequence Models And LSTM Networks https://pytorch.org/tutorials/beginner/nlp/sequence_models_tutorial.html