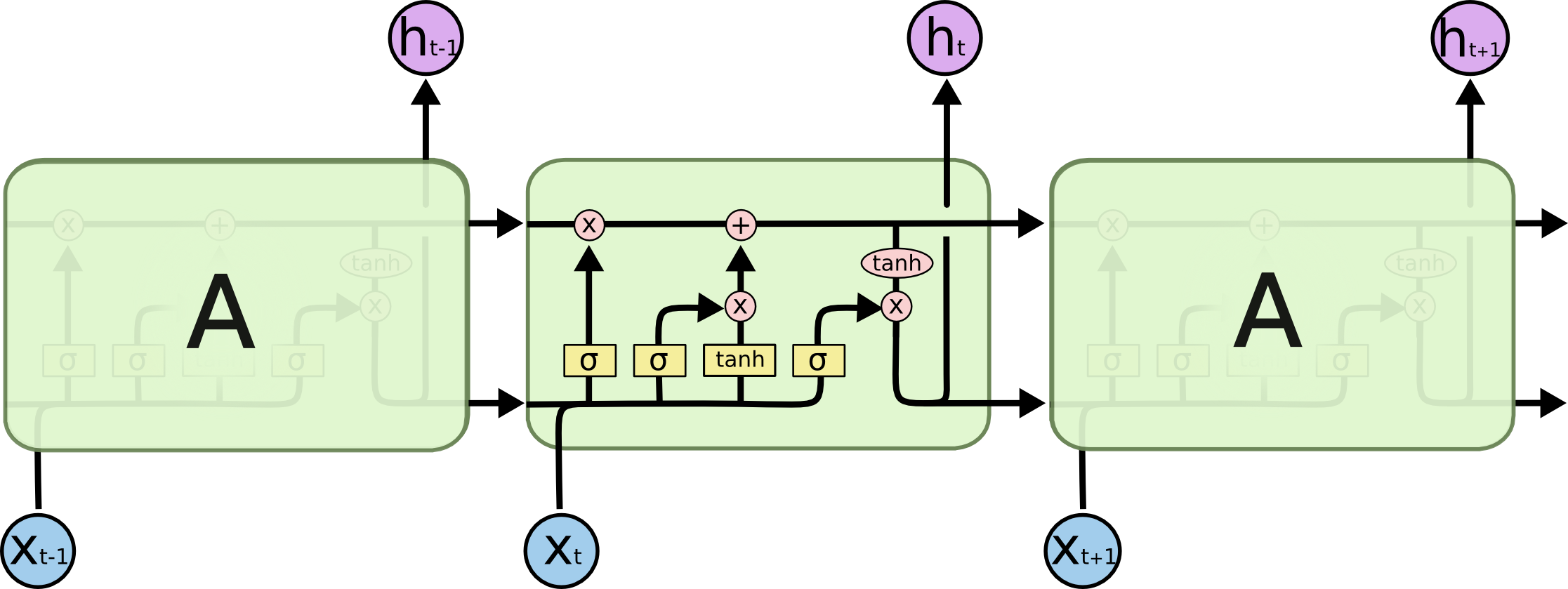
**Data Mining Project**

1. **Dataset**
2. **Related work**
3. **Tasks performed on the datasets**

**3.1 Methodology**



**3.2 Results:**

**RNN:**

**Experimental Setup:**

* Used keras open source library
* Train dataset indexing
* without embedding just indexing (30 epochs, optimizer=adam, drop\_out =0.5) - 77.1 (F1 Measure)

With google 300 dimensions word embeddings (30 epochs, optimizer=adam, drop\_out =0.5)- 79.5 (F1 Measure)

With google 300 dimensions word embeddings (100 epochs, optimizer=adam, drop\_out =0.5, early stopping=10 epochs)- 76.5 (F1 Measure)

With google 300 dimensions word embeddings (100 epochs, optimizer=adam, drop\_out =0.5, early stopping=20 epochs)- 76.36 (F1 Measure)

We performed LSTM using the following settings

|  |  |
| --- | --- |
| Feature Name | Feature value |
| Embedding (X values) | Google word2vec 300 dimensions |
| Indexing (y Values) | One hot encodings |
| Nodes | 128 |
| Dropout | 0.2 to 0.6 |
| Recurrent dropout | 0.2 to 0.8 |
| Model | LSTM |
| Final Layer | Sigmoid |
| Loss | Binary Cross entrophy |
| Optimizer | Adam |
| Metrics | Accuracy |
| Early stopping | Loss on validation with 30 patience |
| Batch Size | 32 |
| Epochs | 1500 |

|  |  |
| --- | --- |
| Dropout | Accuracy |
| 0.2 | 77.40 |
| 0.4 | 78.66 |
| 0.5 | 79.78 |
| 0.6 | 78.13 |

We performed the experiments with varying recurrent dropouts and with the best performing setting of dropout (0.5)

|  |  |
| --- | --- |
| Recurrent Dropout | Accuracy |
| 0.2 | 80.03 |
| 0.4 | 80.59 |
| 0.6 | 61.13 |
| 0.8 | 74.26 |

The following experiments yield the below results based on the above training weights

|  |  |
| --- | --- |
| Dropout | Accuracy |
| 0.2 | 77.26 |
| 0.4 | 78.43 |
| 0.5 | 77.26 |
| 0.6 | 76.84 |

|  |  |
| --- | --- |
| Recurrent Dropout | Accuracy |
| 0.2 | 76.51 |
| 0.4 | 78.57 |
| 0.6 | 78.43 |
| 0.8 | 76.84 |

RNN:

For experiments,

For experimenting we have used the following parameters, we provided two types of results which includes using word2vec and using LSTM rand

We used one hot encodings for output predictions, LSTM is used with 128 nodes, initial layer is either embedding layer for if word2vec is used or random embedding if LSTM rand is used, Adam optimizer is used with loss function of binary cross entropy, the final layer is sigmoid function, we also used early stopping on loss of development set with a patience of 30 epochs, we ran the model on 1500 epochs.

We conducted experiments by changing the dropouts ranging from 0.2 to 0.6 and we found out that with 0.5 the model gave best results. We extended the experiments on recurrent dropout ranging from 0.2 to 0.8, where the model performed best with recurrent dropout of 0.4.

1. **Conclusion**
2. **References**

<http://colah.github.io/posts/2015-08-Understanding-LSTMs/>

<http://papers.nips.cc/paper/5949-semi-supervised-sequence-learning.pdf>

<http://stackoverflow.com/questions/34518656/how-to-interpret-loss-and-accuracy-for-a-machine-learning-model>