Predicting the Occurrence of World News Events using Recurrent Neural Networks and Auto-Regressive Moving Average Models

Emmanuel M. Smith[1], Jim Smith[1], Phil Legg[1], and Simon Francis[2]

[1] University of the West of England, Bristol, United Kingdom
{sinclair.smith, james.smith, phil.legg}@uwe.ac.uk

[2] Montvieux Limited, Gloucestershire, United Kingdom
simon.francis@montvieux.com

Abstract. The ability to predict future states is fundamental for a wide variety of applications, from weather forecasting to stock market analysis. Understanding the related data attributes that can influence changes in time series is a challenging task that is critical for making accurate predictions. One particular application of key interest is understanding the factors that relate to the occurrence of global activities from online world news reports. Being able to understand why particular types of events may occur, such as violence and peace, could play a vital role in better protecting and understanding our global society. In this work, we explore the concept of predicting the occurrence of world news events, making use of Global Database of Events, Language and Tone online news aggregation source. We compare traditional Auto-Regressive Moving Average models with more recent deep learning strategies using Long Short-Term Memory Recurrent Neural Networks. Our results show that the latter are capable of achieving lower error rates. We also discuss how deep learning methods such as Recurrent Neural Networks have the potential for greater capability to incorporate complex associations of data attributes that may impact the occurrence of future events.

1 Introduction

The ability to predict the nature of upcoming events has a variety of potential applications for protecting and understanding our global society. In recent years, there has been much interest in attempting to predict the occurrence of such events [1,3,14,15,18,19,20]. One of the most challenging aspects is to characterise what previously-observed data may have a causal relationship on future activity. Currently, many researchers make use of the Conflict and Mediation Event Observations (CAMEO) [5] framework for coding event types, where events of different types and by various global actors (e.g., countries) are mapped as a time series. Existing statistical methods such as the Autoregressive Moving Average (ARMA) have been used previously in an attempt to predict the occurrence of different event types [18,20], as have Machine Learning (ML) algorithms such as Bayesian methods and random forest classifiers [1,14].

More recently, Recurrent Neural Networks (RNNs) have been used to improve prediction performance on several sequence-based learning problems [11]. In particular, Long Short-Term Memorys (LSTMs) have been used for time series predictions [2,4,13]. Hence, we are interested to study whether the reported benefits of using RNNs in other domains can be applied for predicting global event data. In this paper, we make the following contributions:

1. We present a comprehensive study that compares the use of RNNs with traditional ARMA models, for time series analysis.
2. We apply RNNs and models from the ARMA family to the problem of predicting the occurrence of world news events. We also investigate the suitability of each method for this task and discuss their potential extendability. We show how the RNN approach provides greater flexibility for incorporating additional information that would support its predictive capabilities.

2 Related Work

As global event data has become more accessible, many studies have attempted to make actionable predictions [1,8,14,15,16,18,19,21,22]. Much of this work has focused on statistical time series techniques, such as Autoregressive Fractionally Integrated Moving Average (ARFIMA) models. For example, Brandt et al. developed Markov switching and Bayesian vector autoregression models for predicting material conflict between Israel and Palestine at inter- and intra-state spacial resolutions [1]. Yonamine employed ARFIMA models to predict material conflict for Afghanistan districts [21]. Yuan forecasted the inter-country relationships of China, again using ARFIMA models [22].

Various ML methods have been applied. Perry [14] reported the use of naive Bayes and random forest classifiers on the Armed Conflict Location and Event Dataset (ACLED) event database [17]. Qiao et al. developed a Hidden Markov Model (HMM) for predicting social unrest in five separate south-east Asian countries using Global Database of Events, Location, and Tone (GDELT) data [16]. Phua et al. utilised decision trees to predict the Singapore stock market's Straits Times Index [15], again using GDELT data.

With the exception of HMMs, one limitation is that there is little consideration for the temporal dependence of future events. In other domains such as language modelling and video analysis, LSTMs have been shown to maintain temporality in sequential-based learning tasks [7,12]. Lipton et al. also used LSTMs to perform classification on multivariate time series data, obtained from patients' medical sensor readings [13]. They found that LSTMs perform better than their baseline models, and that heavy use of dropout allows for larger networks which achieve greater results. These clinical events are similar to global events, as they have associated time stamps, actors, and event types. With the recent successes of LSTMs, our study will address how such techniques can be utilised to better characterise and predict the occurrence of global events.

3 Auto-Regressive Moving Average

ARMA models are one of the most commonly used statistical techniques for modeling and predicting time series data [10] by parameterising and combining a number of independent components. The Auto-Regressive (AR) component also uses a weighted linear combination of the past values of the series, essentially performing a regression of the time series against itself. For a target variable x_t,

$$x_t = c + e_t + \sum_{i=1}^{p} \psi_i x_{t-i} \tag{1}$$

where x_t is the value of the time series at time t, c is some constant, e_t is the error at time t which is assumed to be white noise, p is the number of time lags to consider, and ψ are the parameters.

The Moving Average (MA) component also attempts to predict a target variable using a form of regression, although here it is based on the previous forecast errors. Whilst the errors can not be directly observed due to noise, the MA component can be inverted to be in the form of the AR component if the time series is assumed to be stationary. For a target variable x_t,

$$x_t = c + e_t + \sum_{i=1}^{q} \theta_i e_{t-i} \tag{2}$$

where q is the number of time lags, and θ are the parameters. If the assumption made for inverting from AR to MA holds for a given time series, the ARMA is simply the combination of the two components. For a target variable x_t,

$$x_t = c + e_t + \sum_{i=1}^{p} \psi_i x_{t-i} + \sum_{i=1}^{q} \theta_i e_{t-i} \tag{3}$$

One crucial assumption here is that the time series is stationary. Many prediction tasks deal with time series that do not meet this criterion. It is possible to transform a non-stationary time series into a stationary one by taking the difference between each variable and its predecessor. For example, the first order difference x_t^1 of the original variable x_t is $x_t^1 = x_t - x_{t-1}$, and consequently, x_t^2 is the second order difference. As alluded to, allowing the order to be fractional allows for finer granularity in this approach. For a fractional difference x_t^d,

$$x_t^d = \sum_{i=0}^{\infty} \frac{\prod_{j=0}^{i-1}(d-j)}{i!} x_{t-i} \tag{4}$$

where d is the fractional differencing coefficient. Combining the AR and MA components, and incorporating a fractional differencing component gives us the complete ARFIMA model.

4 Long Short-Term Memory

Traditional Artificial Neural Networks (ANNs) are not designed to incorporate temporal dependencies. RNNs extend this, such that each node incorporates a loop back connection which allows a hidden state to be passed through time. At each time step, the node receives its usual inputs, together with the hidden state obtained from the previous time step, allowing for a long term memory to be incorporated. It was later identified that RNNs suffer from the vanishing gradient problem [9]. LSTM, an extension to the RNN architecture, was developed to overcome this issue [7]. This is achieved through gating mechanisms and a cell activation state, in addition to the existing hidden state, since the network learns when to forget long-term information and when to incorporate new information. Separating the hidden state with the cell activation state also allows for the network to learn to control how much of the cell activation it outputs.

As can be seen in Figure 1, a LSTM node takes, as input, a combination of an input vector, $\mathbf{x}$, and the previous hidden state, $\mathbf{h}$.[3] The LSTM first calculates a new candidate cell activation, $\tilde{\mathbf{c}}$, via a weighted sum of these inputs and a bias, b. The result of this calculation passes through a hyperbolic tangent activation function, as denoted by Equation 5.

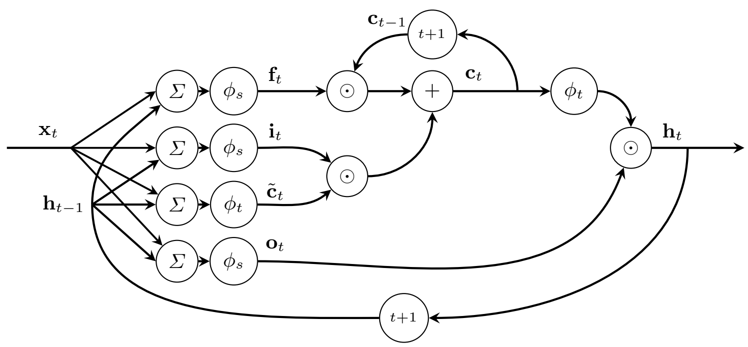

Fig. 1: Structure of a LSTM node, where each arrow represents the movement of vectors, and each circle denotes an operation performed on those vectors. Σ denotes a weighted summation, and $t+1$, denotes a delay of one time step. Input data flows through the node's calculations left-to-right, with the exception of the time delay that is retrieved from the previous time step.

Once the candidate cell activation is determined, the gates control how much of this activation we should keep and how much of cell activation from the previous time step we should forget. The gate that controls how much of the candidate cell activation we should keep is the input gate, $\mathbf{i}$, and the gate that controls how much we forget the past cell activation is the forget gate, $\mathbf{f}$. The final gate, the output gate, $\mathbf{o}$, is incorporated once the hidden state is calculated from the new cell activation. These values are calculated as follows:

[3] Usually the previous cell activation and hidden state are initially set to zero.

$$\tilde{\mathbf{c}}_t = \phi_t(\mathbf{W}_{\tilde{c}}\mathbf{x}_t + \mathbf{U}_{\tilde{c}}\mathbf{h}_{t-1} + b_{\tilde{c}}) \tag{5}$$
$$\mathbf{f}_t = \phi_s(\mathbf{W}_f\mathbf{x}_t + \mathbf{U}_f\mathbf{h}_{t-1} + b_f) \tag{6}$$
$$\mathbf{i}_t = \phi_s(\mathbf{W}_i\mathbf{x}_t + \mathbf{U}_i\mathbf{h}_{t-1} + b_i) \tag{7}$$
$$\mathbf{o}_t = \phi_s(\mathbf{W}_o\mathbf{x}_t + \mathbf{U}_o\mathbf{h}_{t-1} + b_o) \tag{8}$$

where $\tilde{\mathbf{c}}_t$ is the candidate cell activation; $\{\mathbf{f}, \mathbf{i}, \mathbf{o}\}_t$ are the forget, input, and output gate vectors; $\phi_{\{s,t\}}$ are the sigmoid and hyperbolic tangent activation functions; $\{\mathbf{W}, \mathbf{U}\}_{\{\tilde{\mathbf{c}},\mathbf{f},\mathbf{i},\mathbf{o}\}}$ are weight matrices; $\mathbf{x}_t$ is the input vector; $\mathbf{h}_{t-1}$ is the hidden state vector at the previous time step; and $b_{\{\tilde{c},f,i,o\}}$ are biases.

To calculate the final hidden state vector $\mathbf{h}_t$, we must first compute the new cell activation $\mathbf{c}_t$, as a combination of the candidate cell activation, gated by $\mathbf{i}_t$, and the previous cell activation $\mathbf{c}_{t-1}$, gated by $\mathbf{f}_t$. The gating is performed using element-wise multiplication and the activations are then combined using addition. Finally, the hidden state is the cell activation, bounded by the hyperbolic tangent activation function, and gated by $\mathbf{o_t}$, denoted as follows:

$$\mathbf{c}_t = \mathbf{c}_{t-1} \odot \mathbf{f}_t + \tilde{\mathbf{c}}_t \odot \mathbf{i}_t \tag{9}$$
$$\mathbf{h}_t = \phi_t(\mathbf{c}_t) \odot \mathbf{o}_t \tag{10}$$

where $\odot$ is an operator for the element-wise multiplication of vectors.

5 Methodology

In this section, we present a comparative study of using LSTM and ARFIMA models for predicting the occurrence of material conflict events in Afghanistan. We make use of GDELT, which provides a real-time machine-coded data repository of global news event reports. GDELT provides event details such as the actors involved, the severity which ranges from verbal cooperation to material conflict, and a link to the source. Our study thus extends, via LSTMs, the work of Yonamine [21], who originally considered predicting the number of material conflict events that occur in Afghanistan using ARFIMA models.

5.1 Data Representation

GDELT is an event database which automatically collates global news, and identifies and encodes mentioned events along with their associated details. An event usually involves multiple actors which range from countries and known groups, to prominent figures. In an event, an identifiable action is performed at a particular time and location. GDELT was chosen for this research due to its tremendous temporal and geospatial scale, and for its release interval which allows for real-time, actionable predictions.

There are multiple methods of obtaining the GDELT data, however, for the purposes of this study, the dataset was obtained from their BigQuery API. Our query extracted all GDELT 1.0 events, where any of the actor codes contained the substring 'AFG' (Afghanistan), since the year 2000. Even though more historic data is available, we only include data from 2000 onwards in order to cover the time range covered by Yonamine's study [21], namely 2001 to 2012.

The majority of the event features provided by GDELT are categorical, such as the actor, and thus do not easily lend themselves towards time series representations. This makes them unsuitable for use by both models in their standard form. Techniques such as one-hot encoding or other vector representations could be used, however, that is not the main focus of this investigation. After excluding non-numerical features, each event has a date, an event code, an associated latitude and longitude, a Goldstein scale [6], the number of articles mentioning the event, the number of mentions of the event within articles, the number of article sources, and the average article tone. As in Yonamine study, we focus on predicting solely the total count of material conflict events per month from April 2008 to April 2012, given a training set ranging from February 2001.

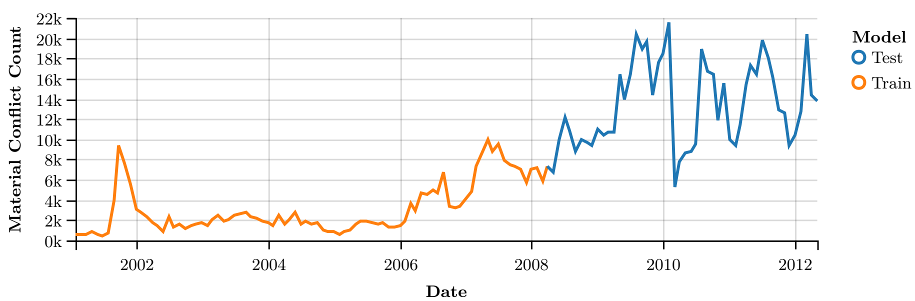

Fig. 2: The number of material conflict events from 2001 to 2012 as recorded by GDELT. The orange segment represents the portion of the time series explicitly used for training, and the blue shows the portion used for model evaluation.

To train both the ARFIMA and LSTM models on the material conflict count per month, the original dataset must first undergo some preprocessing. Firstly, the events must be aggregated temporally, in our case, to the month level. To do this, we filtered the dataset to the desired historical time range and calculated the total count of identified material conflict events for each month, giving us 136 unique data points, the result of which can be seen in Figure 2. When temporally aggregating time series data much consideration is usually warranted, however, for the purpose of this study we focus on the monthly level, as historical global event data research has shown the monthly level to provide better results [21].

5.2 Model Architectures

For the ARFIMA model, we used the automatic parameter estimation functionality provided by the R forecast library, namely the 'arfima' function, which attempts to identify the ideal model parameters using statistical tests. When predicting future values, the function provides confidence intervals and a mean estimate, but for this study, we utilise the mean value.

For LSTM models, there are additional design decisions such as the network size, the activation functions, the initialisation scheme, and regularisation methods. Since we are training the model in a sequence to sequence fashion, the LSTMs were constructed to take as input, a vector consisting of the numerical event features, and predict a vector which consists of the same features but for the next time step. A visualisation of the process in provided in Figure 3.

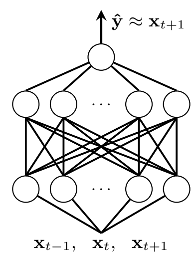

Fig. 3: A diagram of the LSTM model architecture used for this investigation. The network is comprised of two LSTM layers with a fully connected layer.

After some initial experimentation, our network setup contains two LSTM layers with 512 nodes each. We used the standard activation functions and initialisation scheme provided by the 'Keras' library, and added a 50% chance of dropout to each node as regularisation. Since ANNs tend to train more effectively with normalised inputs, the min-max normalisation method was applied to each feature, based on the minimum and maximum values observed in the original training portion of the dataset.

5.3 Training Procedures

To evaluate performance in real world contexts, we devised two distinct training procedures. The first involves training both LSTM and ARFIMA models on the original portion of the data, shown in orange in Figures 2 and 4. When making predictions on the test component of the data we fixed the models' parameters, ensuring that they are not allowed to take into account new information present in the test set. This would be akin to scenarios where the actual target values remain unknown for data update intervals which extend past an actionable time window of prediction.

The second approach is to replicate a real-time predictive scenario, such as that provided by the GDELT data. Again each of the models is initially trained on the training set, but now for the prediction at every subsequent time step in the test set, the models are allowed to adjust their parameters given the additional information (i.e. the orange and red portions of Figure 4).

For each of these training procedures and for all models, at each time step, they are presented with the event features for the current month and are tasked

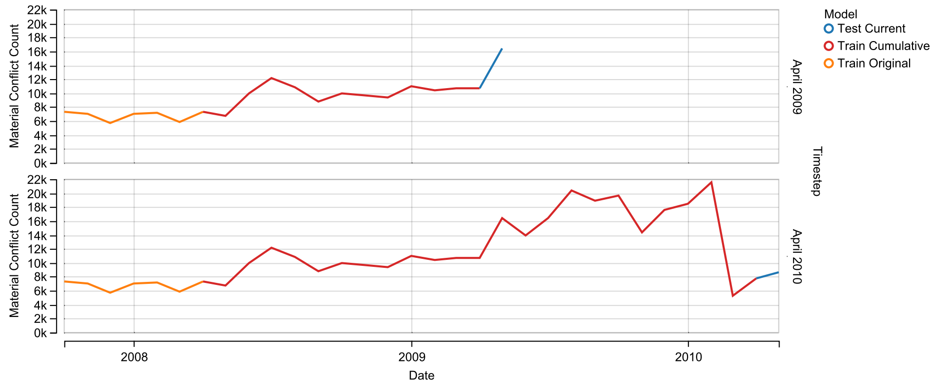

Fig. 4: A demonstration of how additional training data is incorporated into the models at two different time steps. With this 'real-time' setup, for each prediction, the models are allowed to update their parameters using all available historical observations. The red segment shows the additional training data provided to the model, and the blue section shows its relevant prediction target.

with predicting those values for the next month. This provides a solid basis for comparison how ARFIMA and LSTM models perform given different event data contraints. For this investigation, we refer to the first training procedure as the 'fixed' method and the second as the 'real-time' method.

6 Results

As a baseline each of the models were compared against a naive method, which is simply predicting that the next month will be the same as the last. For the sake of clarity, the naive method is omitted from the plots below, as this is simply a delayed version of the actual data.

Figure 5 shows the predicted number of material conflict events in Afghanistan. Each fixed model attempted to make a prediction for the number of material conflict events for the following month without updating its parameters. From Figure 5 we can see that the ARFIMA model matches more closely with the shape of the actual series, however, there seems to be a time lag present in its predictions, similar to the naive approach. The fixed LSTM appears to be capturing a smooth version of the original dataset, however, it fails to properly account for the magnitude changes in the original series.

When allowed to take account of new data both models elicit different behaviours. Figure 6 shows the predictions made by both models when they are allowed to re-estimate their parameters based on the cumulative historic test data. The ARFIMA model no longer overshoots the large drop in early 2010, however, it now overestimates the drop present in early 2011. Overall the real-time ARFIMA model appears to have the same time lag behaviour as the fixed version, however, it appears to be making predictions which diverge further from the naive method.

Fig. 5: Predictions made by ARFIMA and LSTM models on the Afghanistan time series with the fixed training procedure.

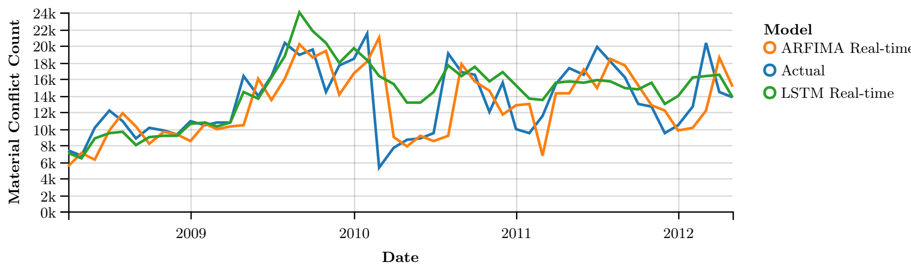

Fig. 6: The predictions made by ARFIMA and LSTM models on the Afghanistan time series using the 'real-time' training approach.

The real-time LSTM model differs significantly from the behaviour of its fixed version. It no longer appears to be a smooth version of the original series, but rather, the model attempts to predict quite significant divergences from the naive method as can be seen in mid-2009. Additionally, the LSTM seems to identify the 2010 downward jump before it happens, however, doesn't account for the magnitude of the change. Allowing the models to use all available historical data seems to have allowed them to make closer predictions, however, it is difficult to tell without evaluating the errors across all points in the test set.

To evaluate these results we used three different commonly used metrics, the Mean Absolute Error (MAE), the Root Mean Squared Error (RMSE), and the Mean Absolute Percentage Error (MAPE) that make different assumptions about how to weight the extent to which a prediction is incorrect. Table 1 provides all three scores for all models tested.

As Figure 2 shows, the magnitude and frequency of variations in the time series violate many of the assumptions of standard hypothesis testing, and in fact, the distribution of error values was typically trimodal, with large positive/negative errors at spikes in the real data. Therefore to compare between two approaches a and b we modelled the distribution of the *differences* between the normalised relative errors, i.e. of $e_{ab}^t = (pred_a^t - pred_b^t)/actual^t$. The null hypothesis is that there is no difference in predictive quality, so the e_{ab}^t are sampled

Model	Method	MAE (σ)	RMSE (σ)	MAPE (σ)
Naive	NA	2632.68 ($\pm$2809.32)	3829.55 ($\pm$2809.32)	22.94 ($\pm$42.57)
ARFIMA	Fixed	2713.66 ($\pm$2747.31)	3841.96 ($\pm$2747.31)	23.09 ($\pm$39.23)
LSTM	Fixed	2674.81 ($\pm$**1997.96**)	3326.65 ($\pm$**1997.96**)	21.70 ($\pm$**22.67**)
ARFIMA	Real-time	2610.70 ($\pm$2774.19)	3789.19 ($\pm$2774.19)	22.47 ($\pm$40.99)
LSTM	Real-time	**2305.31** ($\pm$2208.42)	**3177.11** ($\pm$2208.42)	**21.57** ($\pm$32.79)

Table 1: The error scores for each model and each of the training methods. The lowest error scores for each metric is highlighted in bold for visibility.

from a distribution with mean 0. In most cases e_{ab} was unimodal but skewed, so we used the more conservative non-parametric Wilcoxon signed ranks test to check whether the observed differences are statistically significant.

Comparing to the naive baseline we found that the fixed ARFIMA performed significantly worse ($p < 0.01$) whereas the observed difference between the baseline and fixed LSTM was not significant. Both real-time models predicted significantly more accurately than the baseline ($p < 0.05$). Finally, we found that the real-time LSTM performed significantly better than both the real-time ARFIMA and the fixed LSTM ($p < 0.01$).

7 Discussion

The potential capabilities of LSTMs reach far beyond what we have observed. For one, LSTMs are able to not only predict all provided training features, they can be trained to predict entirely different features, such as the probability of specific types of events occurring.

There are a few limitations of the chosen methodology when it comes to taking advantage of event data. By treating the tasks as a time series problem, we are ignoring categorical features which contain useful event information such as geospatial relationships, e.g. the source and target actors, their ethnic groups, and religions. Using some method of representing these categorical features in a numerical vector space would allow LSTMs to incorporate this additional information. Also, we focus only on the subset of GDELT data that falls under the material conflict QuadClass. Removing the other categories prevents the network from identifying relationships between different types of events, or whether one type of event precedes another. For example, usually verbal conflict events would precede material conflict events. Additionally, the dataset is aggregated to the monthly level, which is common in event data analysis. However, LSTMs have the potential to account for the variance present in the weekly, daily, or even hourly levels. Given that ANNs perform better given more data, training on a fine-grained level of temporal aggregation might improve performance for the LSTM model even though the ARFIMA's would suffer.

Currently, when new data is encountered, the real-time LSTM trains an additional epoch using both the original training data and the new data observed since then. Whilst this provides a promising start, some issues for further investigation remain. Firstly, there is a trade-off between additional training time and the number of real-time training epochs to account for. Also, in a real context, including the entire history would cause the model to progressively become slower. Therefore, a fixed length historical window may likely be required.

One last point of note is that ANNs require some form of normalisation in order to be trained effectively, and LSTMs are no exception. We used a basic min-max normalisation scheme which defined upper and lower bounds for each feature based on the training set, and then future data was normalised accordingly. This choice of normalisation is not ideal as some of the features have large outliers which squash the majority of values to within a small range. Therefore, an improved normalisation scheme would likely provide better results.

8 Conclusion

We have presented a comparative study between LSTM and ARFIMA models for multivariate time series prediction. We show how LSTMs are capable of maintaining long-term memory states that can be informative for modelling changes in the underlying time series, using a case study on material conflict world news events. In both the LSTM and ARFIMA models, we observe a difficulty in identifying the true nature of the time series, however, we also find that the LSTM is more suitable for incorporating additional information that may help give a closer approximation to the underlying data. In our future work, we plan to investigate how more varied attributes can contribute towards the prediction accuracy for LSTMs. We also intend to explore the internal functions of the LSTM to diagnose uncertainties in how the network handles challenging information such as modelling real-world activities.

References

1. Brandt, P.T., Freeman, J.R., Schrodt, P.A.: Real Time, Time Series Forecasting of Inter- and Intra-State Political Conflict. Conflict Management and Peace Science **28**(1), 41–64 (2011). DOI 10.1177/0738894210388125
2. Busseti, E., Osband, I., Wong, S.: Deep Learning for Time Series Modeling. Tech. rep., Stanford (2012)
3. Cadena, J., Korkmaz, G., Kuhlman, C.J., Marathe, A., Ramakrishnan, N., Vullikanti, A.: Forecasting Social Unrest Using Activity Cascades. PLOS ONE **10**(6) (2015). DOI 10.1371/journal.pone.0128879. URL `http://dx.plos.org/10.1371/journal.pone.0128879`
4. Esteban, C., Staeck, O., Yang, Y., Tresp, V.: Predicting Clinical Events by Combining Static and Dynamic Information Using Recurrent Neural Networks (2016). URL `http://arxiv.org/abs/1602.02685`

5. Gerner, D.J., Abu-jabr, R., Schrodt, P.A., Yilmaz, : Conflict and Mediation Event Observations (CAMEO): A New Event Data Framework for the Analysis of Foreign Policy Interactions. International Studies Association, New Orleans (2002). URL http://www.ku.edu/~keds.
6. Goldstein, J.S.: A Conflict-Cooperation Scale for WEIS Events Data. Journal of Conflict Resolution **36**(2), 369–385 (1992). DOI 10.1177/0022002792036002007. URL http://jcr.sagepub.com/cgi/doi/10.1177/0022002792036002007
7. Greff, K., Kumar, R., Koutník, J., Steunebrink, B.R., Schmidhuber, U.: LSTM: A Search Space Odyssey (2015). URL https://arxiv.org/pdf/1503.04069.pdf
8. Hammond, J., Weidmann, N.B.: Using Machine-Coded Event Data for the Micro-Level Study of Political Violence. Research & Politics **1**(2), 1–8 (2014). DOI 10.1177/2053168014539924
9. Hochreiter, S., Bengio, Y., Frasconi, P., Urgen Schmidhuber, J.: Gradient Flow in Recurrent Nets: the Dificulty of Learning Long-Term Dependencies (2001)
10. Hyndman, R.J., Athanasopoulos, G.: Forecasting : principles and practice, online edn. OTexts (2013)
11. Karpathy, A.: The Unreasonable Effectiveness of Recurrent Neural Networks (2015). URL http://karpathy.github.io/2015/05/21/rnn-effectiveness/
12. Karpathy, A., Johnson, J., Fei-Fei, L.: Visualizing and Understanding Recurrent Networks (2015). URL http://arxiv.org/abs/1506.02078
13. Lipton, Z.C., Kale, D.C., Elkan, C., Wetzell, R.: Learning to Diagnose with LSTM Recurrent Neural Networks. International Conference on Learning Representations **4** (2015). URL http://arxiv.org/abs/1511.03677
14. Perry, C.: Machine Learning and Conflict Prediction: A Use Case. Stability: International Journal of Security & Development pp. 1–18 (2013). DOI http://dx.doi.org/10.5334/sta.cr
15. Phua, C., Feng, Y., Ji, J., Soh, T.: Visual and Predictive Analytics on Singapore News: Experiments on GDELT, Wikipedia, and ˆSTI (2014). URL http://arxiv.org/abs/1404.1996http://www.sas.com/singapore
16. Qiao, F., Li, P., Zhang, X., Ding, Z., Cheng, J., Wang, H.: Predicting Social Unrest Events with Hidden Markov Models using GDELT (2013)
17. Raleigh, C., Hegre, H.: Introducing ACLED: An Armed Conflict Location and Event Dataset. In: Disaggregating the Study of Civil War and Transnational Violence (2005)
18. Stoll, R.J., Subramanian, D.: Hubs, Authorities, and Networks: Predicting Conflict Using Events Data. International Studies Association (2006)
19. Weidmann, N.B., Ward, M.D.: Predicting Conflict in Space and Time. Journal of Conflict Resolution **54**(6), 883–901 (2010). DOI 10.1177/0022002710371669
20. Yonamine, J.E.: A Nuanced Study of Political Conflict Using the Global Datasets of Events Location and Tone (GDELT) Dataset. Ph.D. thesis, The Pennsylvania State University (2013)
21. Yonamine, J.E.: Predicting Future Levels of Violence in Afghanistan Districts (2013). URL http://jayyonamine.com/wp-content/uploads/2013/03/Forecasting_Afghanistan.pdf
22. Yuan, Y.: Modeling Inter-Country Connection from Geotagged News Reports: A Time-Series Analysis (2016). URL http://arxiv.org/abs/1604.03647