

11th International Young Scientist Conference on Computational Science

# A study of the influence of news reports and other contextual open-source information on the consumer behavior of bank card users

Grigoryev Mikhail<sup>a,\*</sup>

<sup>a</sup>ITMO University, Kronverksky Pr. 49, bldg. A, St. Petersburg, 197101, Russia

---

## Abstract

We live in a world filled with media. Thus, a hypothesis can be formulated that background context such as news and open-source statistics can influence the way we consume goods in various categories. This research is an endeavour into the impact of context information on consumer behavior. The study of such influence is performed as prediction of consumption time series with context as exogenous variables.

© 2022 The Authors. Published by Elsevier B.V.

This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>)  
Peer-review under responsibility of the scientific committee of the 11th International Young Scientist Conference on Computational Science.

Keywords: Machine Learning ; Natural Language Processing ; Topic Analysis ; Autoregressive Models ; ARIMAX

---

## 1. Introduction

## 2. Related work

### 2.1. Topic analysis

The main goal of this part of literature review is the choice of topic analysis methods to be used in this research work for processing news headlines. Thus, the most acknowledged methods ranging from classic LDA to modern BERTopic are looked upon in this section.

Topic analysis is one of the tasks of natural language processing (NLP). It consists of embedding human-legible texts into machine-readable embeddings, dimension reduction and further clustering of the embeddings. Such clusters can be interpreted as topics found in the texts, hence the name "topic analysis". One of the classical methodologies of topic analysis is the "bag-of-words", an approach for viewing documents

---

\* Corresponding author. Tel.: +7-921-592-4920.

E-mail address: [mikegrig@inbox.ru](mailto:mikegrig@inbox.ru)

as unstructured vocabularies of words with different frequencies. It was the foundation for a plethora of commonly used topic analysis tools such as LDA, NMF, Top2Vec and CTM.

Authors of the publication [8] extracted topics from scientific literature via custom two-stage pipelines. The first step in every experiment was to embed the text into machine-interpretable numeric vectors. Then in the second step different algorithms were used to cluster the vectors, validate the extracted clusters and output an array of topics.

For embedding scientific texts several approaches including word2vec, POS2vec, word-position2vec and LDA2vec were used individually as well as combined into the improved word embedding based scheme. The latter model takes scientific abstract sentences as input, which is processed by word2vec, a simple embedder that transforms all words within the sentence into vectors without taking the word position into account. Then the authors computed averages of word vectors within each sentence so that the resulting vectors could be used as sentence representations. Before averaging, stop-words have been removed due to this tactic yielding better results. After that, POS2vec (part-of-speech2vec) was used on each word in the sentence to generate POS tags. Then, word-position2vec was employed to generate position vectors for the words and the whole sentence. The latter was concatenated to the vectors obtained earlier. Lastly, the sentences were further processed by LDA2vec. Its output vector was concatenated to yield the final ensemble embedding.

In the publication K-means, K-modes, K-means++, SOM and DIANA were chosen as the clustering algorithms. Classical K-means is a simple partition-based clustering method, while K-modes is its extension better suited for categorical data. As K-means' performance highly depends on the initial random seed for the coordinates of cluster centers, K-means++ uses a heuristic function to determine the optimal initial cluster centers. Self-organizing maps (SOM) method is based on clustering and dimension reduction. Divisive analysis clustering (DIANA) is a divisive hierarchical clustering method. Finally, the authors proposed iterative voting consensus (IVC) – a consensus function which seeks cohesion between different clustering methods on the same data and thus lets those algorithms compensate for each others' weaknesses.

The authors have compared 43 different configurations of topic analysis pipeline on the same dataset of scientific abstracts. The models were compared by values of the following metrics: Jaccard Coefficient (JC), FM and F1.

$$JC = \frac{a}{a + b + c} \quad (1)$$

$$FM = \left( \frac{a}{a + b} \cdot \frac{a}{a + c} \right)^{0.5} \quad (2)$$

$$F1 = \frac{2a^2}{2a^2 + ac + ab} \quad (3)$$

Here  $a$  denotes the number of pairs of instances which are grouped in the same cluster and fall in the same category in the initial dataset. The value  $b$  stands for pairs from the same category which were not grouped in the same cluster. Lastly,  $c$  denotes pairs from different categories which were grouped together. The higher the metrics described above, the better results models produce.

According to the results presented in the article and in figure 1 ([8]), among single method clustering strategies, LDA (latent Dirichlet allocation) outperformed other algorithms, with DIANA taking the second place. Results of clustering seemed to have improved significantly when integrating LDA2vec embedding scheme with clustering algorithms.

Even better results were shown by the proposed clustering ensemble framework which utilizes IVC for combining the mentioned methods. Thus, such proposed ensemble approach can lead to better topic analysis models.

As opposed to the example presented above, topic analysis can solve not only the problem of extracting broad semantic themes from texts but also some more nuanced information. Authors of the article [5] propose their two-step approach for aspect-based sentiment analysis. Their procedure lets evaluate reviews in terms of how positive or negative they are and also define aspects of the review object which are evaluated. In the publication, restaurant and laptop reviews were used as input data. The proposed model outputs a tuple in

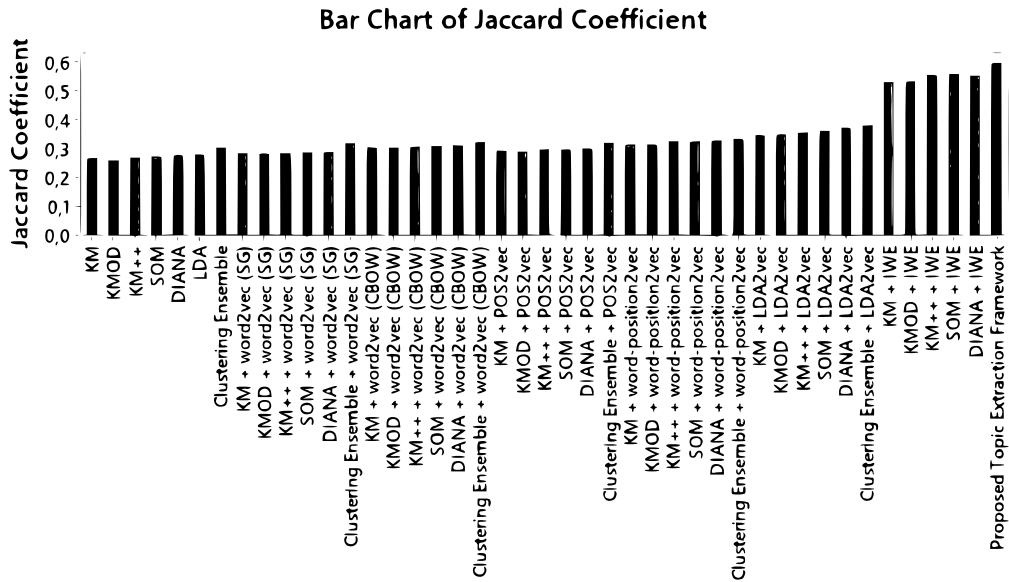


Fig. 1. Jaccard Coefficient metric for different combinations of embedding and clustering methods.

form of (sentiment, aspect). For example, a review of a restaurant could output (good, ambiance) or (bad, food).

For this finesse task a two-step procedure called JASen was developed. It consists of sentence embedding and classification via convolutional neural networks (CNNs). The former takes labeled data for embedding training and unlabeled data for further classification.

The results shown by the JASen model were compared with those of previously developed approaches. Values of aspect identification and sentiment polarity classification are presented in the tables 1 and 2 below ([5]).

Table 1. Quantitative evaluation on aspect identification (%).

Methods	Restaurants				Laptops			
	Accuracy	Precision	Recall	macro-F1	Accuracy	Precision	Recall	macro-F1
CosSim	61.43	50.12	50.26	42.31	53.84	58.79	54.64	52.18
W2VLDA	70.75	58.82	57.44	51.40	64.94	67.78	65.79	63.44
BERT	72.98	58.20	74.63	55.72	67.52	68.26	67.29	65.45
JASen	83.83	64.73	72.95	66.28	71.01	69.55	71.31	69.69

Table 2. Quantitative evaluation on sentiment polarity classification (%).

Methods	Restaurants				Laptops			
	Accuracy	Precision	Recall	macro-F1	Accuracy	Precision	Recall	macro-F1
CosSim	70.14	74.72	61.26	59.89	68.73	69.91	68.95	68.41
W2VLDA	74.32	75.66	70.52	67.23	71.06	71.62	71.37	71.22
BERT	77.48	77.62	73.95	73.82	69.71	70.10	70.26	70.08
JASen	81.96	82.85	78.11	79.44	74.59	74.69	74.65	74.59

In conclusion, the authors have proposed a model fine-tuned for joint aspect-sentiment extraction which outperformed earlier methods for this task.

However, more often the extraction of topics is enough. In contrast to the ensemble approach proposed in the publication [8], authors of the article [3] leverage different modified algorithms that compose BERTopic – a modern and efficient topic analysis tool.

Its defining feature is the ability to extract topics dynamically and to build topic popularity time series. This is allowed by using c-TF-IDF (class term frequency inverse document frequency) representations of topics:

$$\text{c\_TF\_IDF} = \underbrace{\frac{n_t}{\sum_{k \in \text{class}} n_k}}_{\text{c\_TF}} \cdot \underbrace{\log \left[ \frac{|D|}{|\{d_i \in D | t \in d_i\}|} \right]}_{\text{IDF}} \quad (4)$$

Here c-TF is defined by the number of instances of word  $t$  divided by the number of words in the class,  $|D|$  – number of documents in the dataset,  $|\{d_i \in D | t \in d_i\}|$  – number of documents in dataset, containing the word  $t$ . First, BERTopic is fitted on the entire dataset to create a global view of topics. Using c-TF-IDF is efficient, as IDF is computed globally and only c-TF needs to be computed at each timestep.

In the publication, newly proposed BERTopic is compared to its predecessors: LDA, NMF, top2vec and CTM by means of topic coherence (TC) and topic diversity (TD) metrics on three datasets of news and twitter posts. According to the article, BERTopic has high coherence scores across all datasets. In terms of topic diversity, it is outperformed by CTM.

One of the major strengths of BERTopic is that it has the capability to operate in the multilingual mode. The default embedding model "all-MiniLM-L6-v2" shows both moderately high topic coherence and diversity across all supported languages in addition to it being lightweight.

More detailed comparison of BERTopic with other methods of topic analysis is presented in the publication [2]. The authors have chosen latent Dirichlet allocation (LDA), non-negative matrix factorization (NMF), top2vec and BERTopic in the task of topic analysis on the dataset of Twitter posts.

LDA is a generative probabilistic model for discrete data, which could be viewed as three-level hierarchical Bayesian clustering algorithm. Each document is represented as a mixture of topics with corresponding probabilities and each topic is a mixture over the collection of topic probabilities.

NMF is a decompositional method which works of TF-IDF transformed data by breaking down the input term-document matrix ( $A$ ) into a pair of lower-ranking matrices:

- terms-topics ( $W$ ) matrix containing basis vectors;
- topics-documents ( $H$ ) matrix containing weights.

In NMF all elements in those matrices are non-negative so as to be interpretable.

Top2vec algorithm uses word embeddings via pretrained embedding models so that semantically close words have spatially close embedding vectors. Due to the sparsity of the vector space, a dimension reduction is performed before clustering. Commonly, hierarchical density-based spatial clustering of applications with noise (HDBSCAN) is used to identify dense regions in the reduced vector space of documents. As words that appear in several documents cannot be assigned to one document, they are recognized as noise. Thus, no lemmatization is needed beforehand.

BERTopic uses BERT pretrained embedders alongside with sentence-transformers for turning documents into vectors across more than 50 languages. Similarly to top2vec, BERTopic uses uniform manifold approximation and projection (UMAP) for dimension reduction and HDBSCAN for clustering. The principal difference between BERTopic and top2vec is that the latter utilizes c-TF-IDF metric instead of normal TF-IDF. Thus, importance of words in clusters and not in documents is taken into account. The usage of HDBSCAN eliminates the need for lemmatization.

Human interpretation of the extracted topics has shown that BERTopic and NMF perform the best among four tested methods. Despite that top2vec and BERTopic both use pretrained embedders for the

representation of documents in vector space, many topics extracted by top2vec contained several semantic concepts or intersected each other. LDA gave the least legible results of all methods.

The authors mentioned several drawbacks of BERTopic models such as them yielding large numbers of clusters which leads to the need in manual topic processing. Another mentioned disadvantage is that one document could potentially be assigned to just one topic, which often does not reflect reality. In spite of this, BERTopic still can be characterized as one of the most successful and universally applicable methods of topic analysis due to high model quality and efficiency. Thus, BERTopic was the topic analysis method of choice in the current research work.

## 2.2. Time series prediction

Time series forecasting is one of the most important tasks related to time-dependent processes. A plethora of methods for predicting time series have been developed and the most practical ones were described in the publication [9]. The authors group all methods into two major categories: parametric and non-parametric.

Among the parametric methods the following were described:

1. Moving averages models (MA), simple models that view predicted values at a certain timestep as averages over some interval before this step. MA model is defined by the equation:

$$z_{t+1} = \frac{\sum_{i=0}^{r-1} z_{t-i}}{r} \quad (5)$$

Here,  $t$  is the timestep,  $z_{t+1}$  is the predicted value and  $r$  is the number of observations included in the average. Such models are simple but lack quality, when data has seasonality, trend or high-frequency noise.

2. Simple exponential smoothing models (SES) are conceptually close to MA. The main difference is that all previous timesteps are taken into account with different weights, which exponentially decrease the further away from prediction the timestep is. They are defined by the following equation:

$$z_{t+1} = L_t = \sum_{i=0}^{t-1} \alpha(1-\alpha)^i z_{t-i} \quad (6)$$

Here  $\alpha \in (0, 1)$  is the weight for constant smoothing and  $L_t$  is the estimate for the next step at time  $t$ . For ease of computation, recurrent simplification can be formalized:

$$L_t = \alpha z_t + (1-\alpha)L_{t-1} \quad (7)$$

Initially, it is supposed that  $L_1 = z_1$ . The main drawback of the method is the difficulty in optimizing  $\alpha$  and inaccurate results when dealing with trends.

3. Holt's exponential smoothing (HES) is an extension of SES that utilizes a second smoothing constant  $\beta$  for modeling trends. HES models are defined by equations:

$$L_t = \alpha z_t + (1-\alpha)(L_{t-1} + T_{t-1}) \quad (8)$$

$$T_t = \beta(L_t - L_{t-1}) + (1-\beta)T_{t-1} \quad (9)$$

$$z_{t+h} = L_t + hT_t \quad (10)$$

Here,  $L$  and  $T$  are the level and trend components, accordingly, and  $h$  is the prediction horizon (more than 1 step as opposed to SES). Initially,  $L_1 = z_1$  and  $T_1 = z_2 - z_1$ . As in SES method, the most difficult part is the optimization of  $\alpha$  and  $\beta$  constants.

4. Holt-Winters' seasonal exponential smoothing models (HW) iterate on HES method principles and additionally deal with seasonality by adding another constant  $\gamma$ . More commonly used additive HW models (AHW) are defined by equations:

$$L_t = \alpha(z_t - S_{t-s}) + (1 - \alpha)(L_{t-1} + T_{t-1}) \quad (11)$$

$$T_t = \beta(L_t - L_{t-1}) + (1 - \beta)T_{t-1} \quad (12)$$

$$S_t = \gamma(z_t - L_t) + (1 - \gamma)S_{t-s} \quad (13)$$

$$z_{t+h} = L_t + hT_t + S_{t-s+h} \quad (14)$$

Here,  $S$  is the seasonal component and  $s$  denotes the number of timesteps that make a full seasonal cycle.  $L$  is usually initialized with:

$$L_s = \frac{1}{s} \sum_{i=1}^s z_i \quad (15)$$

$T$  with:

$$T_s = \frac{1}{s} \sum_{i=1}^s \frac{z_{s+i} - z_i}{s} \quad (16)$$

and  $S$  indexes are computed:

$$S_i = z_i - L_s, \quad i \in [1, s] \quad (17)$$

5. (S)ARIMA, (seasonal) autoregressive integrated moving average models provide even higher quality forecasts for stochastic time series and conceptually include three operations: autoregression AR with parameter  $p$ , intergration I with parameter  $d$  and moving averages MA with parameter  $q$ . Here, intergration refers to taking successive differences from the time series ( $\Delta z_t = z_t - z_{t-1}$ ) so as to make the time series stationary, i.e. with no trend and constant deviation. ARIMA of order  $(p, d, q)$  is defined by the equation:

$$I'_t = \Delta^d z_t = \delta + \sum_{i=1}^p \varphi_i I'_{t-i} + \sum_{i=1}^q \theta_i e_{t-i} + e_t \quad (18)$$

Here,  $I'_t$  is the value of the  $d$ -times differenced stationary time series,  $\varphi_i$  are AR parameters with lags up to  $p$  and  $\theta_i$  are MA parameters with lags up to  $q$ . Parameter  $\delta$  is the initial level of the model and the last term  $e_t$  denotes white noise with zero mean and constant deviation. If  $d > 1$ , the constant  $\delta$  can be omitted. SARIMA model is an extension of ARIMA that takes in four additional parameters and models seasonality. It is essentially ARIMA  $(p, d, q)$  with a seasonal part added. SARIMA of order  $(p, d, q) \times (P, D, Q)_s$ , where  $P$ ,  $D$  and  $Q$  are maximum lags of AR, differencing degree and maximum lags of MA of the seasonal component denoted as  $I''_t$  below:

$$I_t = \underbrace{\delta + \sum_{i=1}^p \varphi_i I'_{t-i} + \sum_{i=1}^q \theta_i e_{t-i} + e_t}_{\text{ARIMA}} + \underbrace{\sum_{i=1}^P \Phi_i I''_{t-is} + \sum_{i=1}^Q \Theta_i e_{t-is}}_{I''_t = \Delta^D z_t} \quad (19)$$

As in ARIMA, here  $\delta$  can be omitted if  $d + D > 1$ . (S)ARIMA models are the most commonly used methods of time series forecasting in the modern practice due to algorithm efficiency, ease of use and high quality of predictions on a large range of time series. A detailed usage example of ARIMA on the

COVID-19 epidemic dataset is presented in the publication [1]. In the article, as ARIMA is a widely-used model utilized for time series forecasting, no testing of the model was presented. The authors predicted values for the differenced COVID-19 dataset and provided confidence intervals for the forecast. This, as well as the autocorrelation analysis before model fitting gives insight into the real-world usage of ARIMA models on actual data, which provides estimations for the future.

In addition, the authors gave a brief description of non-parametric methods for forecasting processes, which include the following:

1. Artificial neural networks (ANNs) are models comprised of neurons – objects capable of taking several inputs, combining them linearly with a certain bias and feeding the output to an activation function (often sigmoid or relu). The output of a single neuron can be described by the equation:

$$f(\text{net}) = f\left(\sum_{i=1}^l w_i x_i + b\right) \quad (20)$$

Here,  $x_i$  and  $w_i$  are inputs and weights,  $b$  is a bias and  $f$  is the activation function. Arranging neurons in several layers (multiple layer perceptron, MLP) was allowed by introducing backpropagation learning algorithms, where the weights are adjusted to better fit the data in reverse layer order. An example usage of multilayer feed-forward neural networks (MLFFNN) is presented in the publication [6]. The authors showed that even such simple architecture of ANNs can produce good quality forecasts. The paper contains R-metrics (correlation coefficients between predictions and test data) for a dataset of wind speed time series for MLFFNN with values as high as 0.9995. In addition, the authors suggest adaptive neuro-fuzzy inference system (ANFIS) – a method that combines ANNs with fuzzy inference system. The latter implies the usage of fuzzy-parameters (if-else rules manually trained by specialists in the research field). According to the results presented in the article, the utilization of such rules slightly improves time series prediction as opposed to MLFFNN. In the networks mentioned above, neuron-to-neuron signals flow only from input to output. This does not apply to recursive neural networks (RNNs), where neurons form a cycle and the signal has several flow directions. In the simple recurrent network (SRN), the state of a chosen layer within a cycle is conditioned by a context layer on its previous state. This creates short-term memory, i.e. the ability of the network to store complex signals. The most advanced RNNs for forecasting time series are the long short-term memory (LSTM) networks. LSTMs are described by the authors of the publication [4] as essentially RNNs tuned for learning long-term dependencies via components called "memory blocks". The latter are recurrent subnets comprised of memory cells and gates. Memory cells' role is to remember the current state of the network and the gates are in charge of controlling the flow of signal in the network. By their role, gates are classified as input (control the amount of new input data that flows into the network), output (the amount of information that flows out of the memory block) and forget gates (decide the amount of information that remains in the current cell). The layout of a memory block is presented in figure 2 ([4]). It is first decided, which old information stays in the cell, then new information is chosen to be stored. Updating of the cell state is performed by multiplying the old memory state by the forget gate output. Then the element-wise product of the input gate and the candidate values for the new information is added. This procedure lets LSTMs predict time-series with long-term dependencies. Several architectures of LSTM models exist that differ by how the neurons in each memory block are connected. As opposed to fully-connected neurons in the traditional LSTM, they are randomly connected in randomly connected LSTMs (RCLSTM), proposed by the authors of the paper [4]. This architecture more closely resembles real synapses in the human brain, which was one of the reasons for the proposal. The authors tested RCLSTM on several datasets including a dataset of traffic information. The influence of the percentage of connected neurons on the predictive quality was estimated by comparison of RMSE metrics. RCLSTM took less computational resources than standard LSTM, however seemed to be outperformed by it. Taking the above into account, RCLSTM still showed better predictions than commonly used

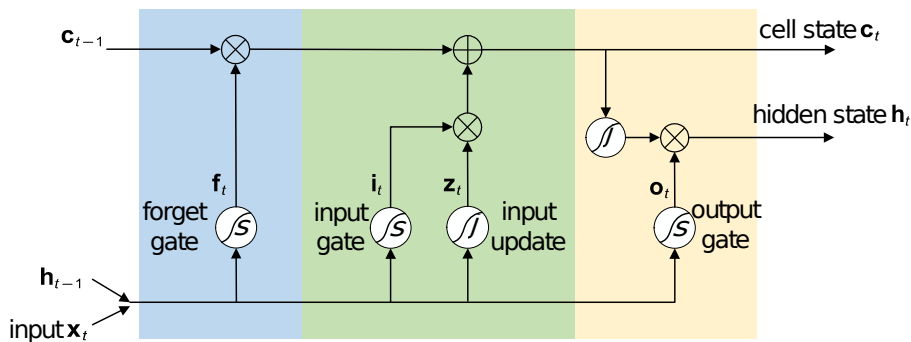


Fig. 2. An illustration of LSTM memory block.

methods such as SVR (support vector machines), ARIMA and FFNN (feed-forward neural networks). This implies that random connection of neurons within LSTMs lets significantly reduce computing time while slightly compromising quality. Even more structurally complicated versions of LSTMs exist. The authors of [7] suggested GeoMAN ("multi-level attention network to predict the the readings of a geo-sensor..."), that is tuned for spatio-temporal correlations. In the paper the authors test this network on a set of geosensors, each providing time-dependent signals (time series). The network utilizes the multi-level attention mechanism including local and global spatial attention as well as temporal attention. This lets the network extract both correlations between different geosensors and also within one time series. Additionally, the authors implemented external factor fusion module, a mechanism of including exogenous variables from different domains for better predictions. The actual architecture of the GeoMAN model is depicted in the figure 3 ([7]).

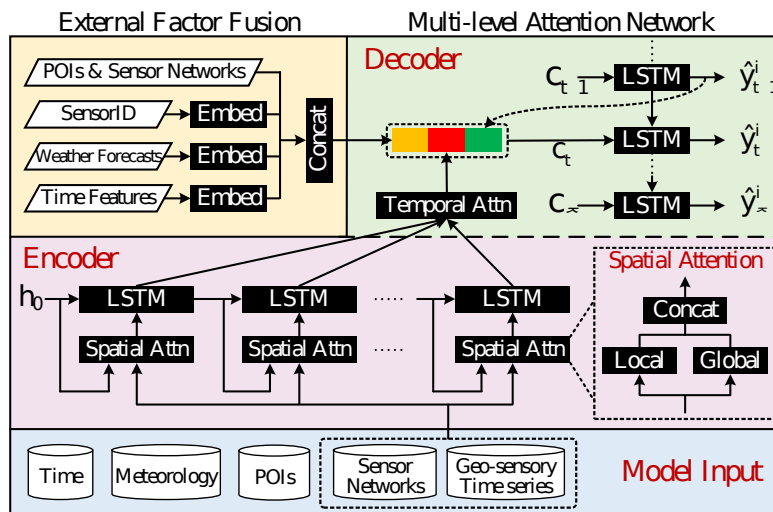


Fig. 3. The framework of GeoMAN.

It can be essentially broken down to two LSTM models, one acting as an encoder for spatial attention and the other as a decoder for temporal. The model was tested on spatio-temporal datasets of water and air quality and within the experiment it was also compared to nine commonly used baseline models including



ARIMA, VAR (vector auto-regressive) and many variations of RNNs. According to the publication, on spatio-temporal data GeoMAN outperforms all baseline models, both parametric and non-parametric, judging by RMSE values. Thus, the paper [7] proves that RNN models can be effectively engineered to perform well on spatio-temporal data.

2. Support vector machines (SVMs) are algorithms conceptually close to ANNs but minimizing the training error while also minimizing the upper bound on the error when the model is applied to test data. The classical illustration to SVMs, binary classification, is presented in figure 4 ([9]).

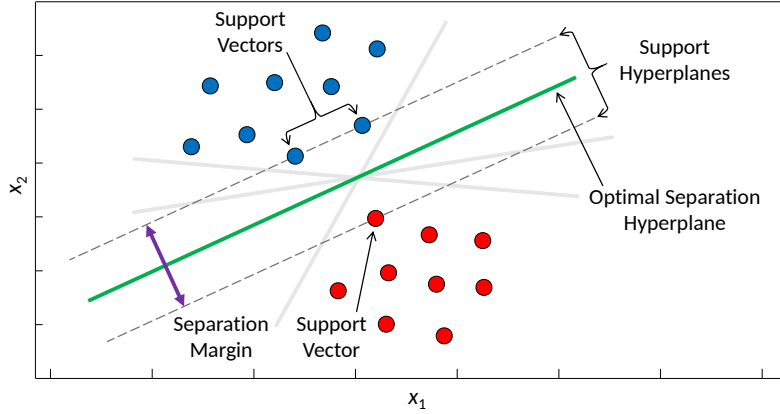


Fig. 4. Binary classification using support vector machines.

The figure shows a set of hyperplanes dividing the two classes. An SVM seeks the optimal divider, where the highest separation margin (distance from divider to support vectors) is reached. As a part of a broader research, SVMs are used along other models in the publication [6] for predicting wind speeds. It showed highly accurate predictions with  $R$ -statistics of up to 0.9938, which is accurate, albeit less accurate than the other methods that comprised the research in [6].

3.  $k$ -nearest neighbors ( $k$ NN) are in nature similarity-based classification algorithms. For forecasting  $z_{m+1}$  for a time series  $Z = (z_1, \dots, z_m)$  the algorithm uses last  $l$  timesteps as query  $Q$  and searches for  $k$  most similar subsequences to  $Q$  by sliding a window of size  $l$  along the time series. Then, the values of  $S_{l+1}^{(j)}$  are averaged in an ensemble to predict  $z_{m+1}$ :

$$f(S) = \frac{1}{k} \sum_{j=1}^k S_{l+1}^{(j)} \quad (21)$$

Such  $k$ NN models have shown the capability to model highly complex non-linear patterns, especially in form of time series prediction with invariance  $k$ NN ( $k$ NN-TSPI). The later algorithm deals with trivial matches, amplitude, offset and complexity invariance.

4. The authors of the publication [10] offered a hybrid method that combined state space models (SSMs) with deep neural networks. Principally, SSMs utilize a latent space that is used to encode time series' features: level, trend and seasonality. Linear SSMs are described by the equation:

$$\mathbf{l}_t = \mathbf{F}_t \mathbf{l}_{t-1} + \mathbf{g}_t \varepsilon_t, \quad \varepsilon_t \sim \mathcal{N}(0, 1) \quad (22)$$

Here,  $\mathbf{l}_t$  is the state of the latent space at timestep  $t$ ,  $\mathbf{F}_t$  is a deterministic transition matrix and  $\mathbf{g}_t \varepsilon_t$  is a random innovation. The predicted value is defined by the equation:

$$z_t = \mathbf{a}_t^\top \mathbf{l}_{t-1} + b_t + \sigma_t \varepsilon_t, \quad \varepsilon_t \sim \mathcal{N}(0, 1) \quad (23)$$

Here,  $\mathbf{a}^\top \in \mathbb{R}^L$ ,  $\sigma_t \in \mathbb{R}^{>0}$  and  $b_t \in \mathbb{R}$  are time-varying parameters of the model. SSMs in general are optimized for applications where the time series are structured and this structure is well-known. The major drawback of SSMs is that multivariate analysis of several correlated time series is impossible as the model is fitted onto one process at a time. In the publication, a hybrid model is proposed, that takes advantage of the benefits of both SSMs and RNNs. The model is comprised of an RNN that is used to parametrize a particular linear SSM. RNN's parameters are learned simultaneously from the whole dataset with all additional covariates, which lets the extraction of shared patterns. This "orchestration" of several SSMs with an RNN is a complicated model architecture, which was quantitatively tested by the authors of the paper [10] against more conventional baseline models such as ARIMA and DeepAR, an RNN-based method. The models were used to forecast electricity consumption and traffic occupancy rates. They were compared by the values of standard  $p50$  and  $p90$  quantile loss metrics. On the one hand, the majority of the tests with forecasting long-term horizons showed that the proposed hybrid method slightly outperformed the others. On the other hand, DeepAR – RNN – performed better at predicting a short forecast horizon.

### 2.3. Prediction with exogenous variables

### 2.4. Impact of context information on consumption

## 3. Conclusion

## Acknowledgements

## References

- [1] Benvenuto, D., Giovanetti, M., Vassallo, L., Angeletti, S., Ciccozzi, M., 2020. Application of the arima model on the covid-2019 epidemic dataset. *Data in brief* 29, 105340.
- [2] Egger, R., Yu, J., 2022. A topic modeling comparison between lda, nmf, top2vec, and bertopic to demystify twitter posts. *Frontiers in Sociology* 7.
- [3] Grootendorst, M., 2022. Bertopic: Neural topic modeling with a class-based tf-idf procedure. *arXiv preprint arXiv:2203.05794*.
- [4] Hua, Y., Zhao, Z., Li, R., Chen, X., Liu, Z., Zhang, H., 2019. Deep learning with long short-term memory for time series prediction. *IEEE Communications Magazine* 57, 114–119.
- [5] Huang, J., Meng, Y., Guo, F., Ji, H., Han, J., 2020. Weakly-supervised aspect-based sentiment analysis via joint aspect-sentiment topic embedding. *arXiv preprint arXiv:2010.06705*.
- [6] Khosravi, A., Machado, L., Nunes, R., 2018. Time-series prediction of wind speed using machine learning algorithms: A case study osorio wind farm, brazil. *Applied Energy* 224, 550–566.
- [7] Liang, Y., Ke, S., Zhang, J., Yi, X., Zheng, Y., 2018. Geoman: Multi-level attention networks for geo-sensory time series prediction., in: *IJCAI*, pp. 3428–3434.
- [8] Onan, A., 2019. Two-stage topic extraction model for bibliometric data analysis based on word embeddings and clustering. *IEEE Access* 7, 145614–145633.
- [9] Parmezan, A.R.S., Souza, V.M., Batista, G.E., 2019. Evaluation of statistical and machine learning models for time series prediction: Identifying the state-of-the-art and the best conditions for the use of each model. *Information sciences* 484, 302–337.
- [10] Rangapuram, S.S., Seeger, M.W., Gasthaus, J., Stella, L., Wang, Y., Januschowski, T., 2018. Deep state space models for time series forecasting. *Advances in neural information processing systems* 31.