StyleLTC: Enhancing Claim Detection with Stylistic Features

Anonymous ACL submission

Abstract

Claim identification is crucial in NLP for detecting assertive statements, especially with the rise of generative AI and automated fact-checking. Traditional neural networks struggle with the temporal dynamics of language. This paper introduces StyleLTC, which uses liquid neural networks with continuous-time properties to overcome these issues. It also incorporates stylistic features to predict claims. Evaluations show that liquid neural networks outperform static models, offering higher accuracy, robustness, and efficiency. StyleLTC achieves comparable accuracy with only 0.612 MB of memory, far less than traditional models, making it highly scalable and effective for claim detection in combating misinformation.

1 Introduction

The spread of misinformation in digital and scientific communications threatens information accuracy. As digital platform reliance grows, effective claim detection—distinguishing factual from misleading statements—becomes crucial, particularly in specialized domains where traditional methods struggle. Automated systems for large-scale claim detection are necessary for journalists, researchers, and the public to assess statements.

Defining a claim is a non-trivial task, especially in automated fact-checking and argument mining (Alam et al., 2020; Gupta et al., 2021; Nakov et al., 2022; Konstantinovskiy et al., 2021; Panchendrarajan and Zubiaga, 2024). We define factual claims as statements verifiable as true or false based on empirical evidence (Ni et al., 2024). Claim spotting and pledge detection (Arslan et al., 2020; Atanasova et al., 2018; Barron-Cedeno et al., 2020; Subramanian et al., 2019; Fornaciari et al., 2021; Schlichtkrull et al., 2024) have been explored, with tools like ClaimRank (Jaradat et al., 2018) and ClaimBuster (Hassan et al., 2017) aiding in detection. Hybrid models (Kartal and Kutlu, 2022) and neural ranking models (Hansen et al., 2019) also contribute to claim identification. Large language models (LLMs) have been used for automatic claim detection (Qi et al., 2024; Quelle and Bovet, 2024), but their high hardware requirements and the impracticality of fine-tuning or few-shot prompting on small, imbalanced data make them unsuitable for real-world claim identification tasks.

In contrast, Liquid Neural Networks (LNNs)(Lechner et al., 2020; Hasani et al., 2021) represent a significant advancement in modeling sequential data, as they dynamically adapt to temporal patterns, allowing for more effective processing of time-dependent information. The integration of linguistic stylistic features, such as usage of grammatical category of words, sentiment, tone, conviction, and concreteness, is critical for improving the precision and reliability of automated claim identification systems. These stylistic elements provide important contextual and cognitive signals that reflect the degree of certainty, engagement, and intent within the text, which are essential for accurate classification. Traditional models often fail to capture these nuances, leading to limitations in handling complex datasets.

In this paper, we proposed Style-Infused Liquid Neural Network (StyleLTC) that embed stylistic features into the LNN framework to combine the advantages of linguistic feature and temporal dynamics, resulting in enhanced accuracy, robustness, and efficiency. The lightweight architecture, requiring only 0.612 MB of memory, achieves performance comparable to more resource-intensive models while significantly reducing memory usage, making it highly scalable. Furthermore, StyleLTC demonstrates domain extensibility, robustness to skewed data distributions, and superior resource efficiency, positioning it as a powerful and efficient solution for claim identification task in diverse domains.

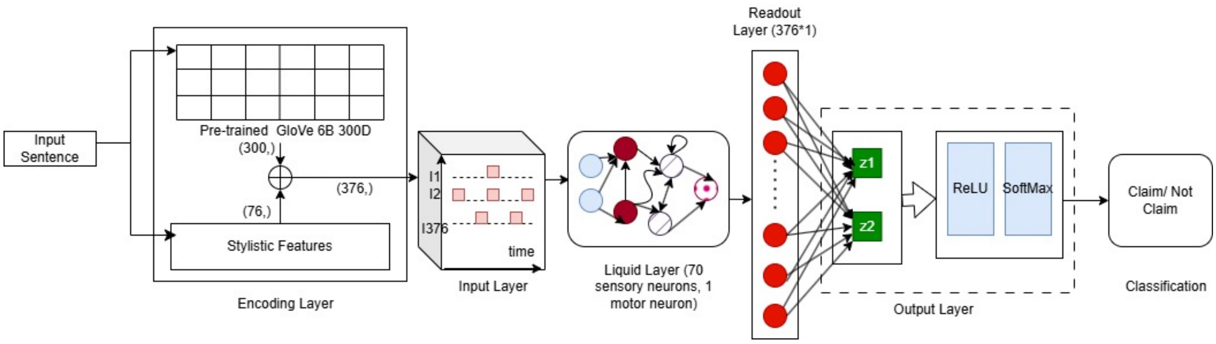

Figure 1: Liquid Time-Constant Network based model architecture for the claim identification.

2 Model Architecture

Hasani et al. introduced the LTC model by modifying the hidden state flow of continuous-time recurrent neural networks (CTRNN) (Funahashi and Nakamura, 1993), defined as: $\frac{dx(t)}{dt} = -\left[\frac{1}{\tau} + f(x(t), I(t), t, \theta)\right] x(t) + f(x(t), I(t), t, \theta)$. Here, $x(t)$ is the hidden state, $I(t)$ the inputs, τ the time-constant, and f is a neural network parameterized by θ and A. LTC networks offer key advantages: 1) an input-dependent time-constant τ_{sys}, introducing the "liquid" nature to RNNs, enabling updates after training; 2) they are universal approximators, able to approximate any autonomous ordinary differential equation(ODE) with finite neurons. For the liquid neural network in our model, we use Neural Circuit Policies (NCP) wiring (Lechner et al., 2020), inspired by biological neural circuits like those in C.elegans. AutoNCP mimics the nematode's nervous system, with four neuron layers: Sensory Neurons, Inter Neurons, Command Neurons, and Motor Neurons.

Our StyleLTC model (Figure 1) for claim identification uses pre-trained GloVe embeddings and LNNs. It has three main components: pre-trained GloVe with stylistic features for text embedding, an LNN for sequence processing, and a fully connected layer for final identification. The process starts with pre-trained GloVe embeddings (Pennington et al., 2014), where each word is mapped to a 300-dimensional vector. Words are tokenized, and their embeddings are retrieved, with unseen words given a zero vector. Sentence embeddings are generated by averaging the word embeddings to represent the sentence's meaning.

The Stylistic Feature Vector Generation process involves creating a 73-dimensional vector using LIWC (Pennebaker et al., 2015), which analyzes the frequency of categories like Nouns, Verbs, and Subject-Verb Agreement in a document generating a vector $[x_1, x_2, ..., x_{73}]$, where each x_i represents the frequency of a specific category. In addition, stylistic features such as *Vagueness*, *Commitment*, and *Conviction* scores are introduced (Sinha et al., 2020). Conviction refers to strong beliefs, that identifies the presence of pride and trust and the absence of timidity, nervousness, or confusion. Commitment is linked to optimism, zest, gain, and achievement in a text. Vagueness reflects a lack of precision or clarity in language, indicating imprecision or ambiguity. Combining these three scores with the LIWC vector forms a 76-dimensional stylistic vector. By concatenating this with the 300-dimensional GloVe vector, we obtain a 376-dimensional style-aware embedding, which is fed into the Input Layer (Ref: Fig 1).

The *input layer* receives a sequence of embedding vectors, typically designed for time-series data. To adapt non-temporal data, the time dimension is set to 1, ensuring compatibility without introducing explicit temporal progression.

The *liquid layer* processes the sequence, capturing temporal dependencies. The sensory and motor neurons are set to 70 and 1 (output size), respectively. Optimal number of sensory neurons was determined based on F1 scores for Political Bias detection in News Articles, as presented in Figure 3 in Appendix B.

In the *readout layer*, the output of the liquid layer is passed through a fully connected layer. This layer map the high-dimensional states to the output, which is then processed by subsequent layers.

The *output* of the fully connected layer is passed through a ReLU activation function and a sigmoid activation function for binary classification. This layer gives the final predicted label for the input sentence.

3 Evaluation

Table 1: Dataset statistics

Dataset	Total Sentences	Claims	Not Claims
Checkworthy English (CT) (Barrón-Cedeño et al., 2024)	23851	5759	18092
Environmental Claim (ECD) (Stammbach et al., 2023)	2647	665	1982
Green-Claims (GCC) (Woloszyn et al., 2021)	773	267	506
Scientific Claim (SCDC) (Achakulvisut et al., 2019)	11519	4000	7519

We evaluated our model's performance using four diverse datasets for an unbiased assessment, with detailed statistics available in Table 1 (See Appendix A for data descriptions). We conducted three experiments based on the datasets. In Experiment-I, each dataset was split into 70% for training and 30% for testing, with the Train and Test sets predetermined for Dataset-CT. For other datasets, a 70:30 split and Monte Carlo cross-validation (Xu and Liang, 2001) were applied, averaging performance metrics to ensure robust generalization. In Experiment-II, we combined all datasets to create an annotated corpus of 38,790 documents, followed by a 70:30 train-test split and Monte Carlo cross-validation. For the SCDC dataset, sentences were grouped into two output classes ("claim" and "not-claim") instead of the original six classes. In Experiment-III, we tested the model's ability to learn from unseen data by training on one dataset (e.g., ECD) and testing on others (GCC).

We used the pre-trained BERT-large model (Vaswani et al., 2017) as our baseline. To train the model, we set the early stopping of training to 800 steps to prevent over-fitting. We use a batch size of 32, a maximum sequence length of 200, and a learning rate of $2 * 10^{-5}$ for training. We have also compared StyleLTC's results with BiLSTM-att, DistilBERT, BERT+style (768-dimensional text embeddings of pre-trained BERT are concatenated with 76 dimensional stylistic features), LTC+GloVe model (when GloVe embeddings are given as input), CFC model (details available in Appendix C) and different open-source LLMs such as LLAMA-3.1 8B and Mistral 7B. LTC+BERT denotes to the instance when 768-dimensional text embeddings of pre-trained BERT given as input to LTC model. In LTC+BERT+style, 768-dimensional text embeddings of pre-trained BERT concatenated with 76 dimensional stylistic features, is given as input.

3.1 Results

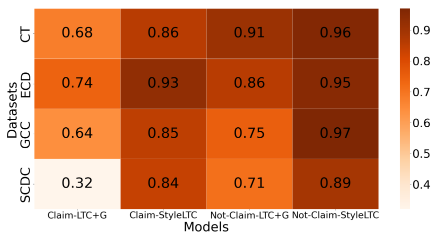

Figure 2: Class-wise F1 Score analysis across different datasets for LTC+Glove and StyleLTC, demonstrating the benefits of stylistic features on claim identification.

Figure 2 demonstrates the benefits of integrating stylistic vectors into text embeddings (LTC+G denotes to LTC+GloVe). As we can observe, StyleLTC shows significantly better performance in identifying both *claim* and *not-claim* sentences than LTC+GloVe. Our results show that while BERT and GloVe embeddings capture contextual nuances, combining them with stylistic factors greatly enhances claim detection. The StyleLTC model (LTC+GloVe+style) outperforms BERT in detecting claims, with improved performance across the CT, ECD, and GCC datasets. For example, the F1 score increases by 41% for Dataset-ECD and 2% for SCDC, while stylistic features improve the F1 score by 19% in Dataset-GCC. StyleLTC performs better than BERT and BiLSTM-att, even with only 100,000 parameters compared to BERT's 110 million. Though BERT-based models are more accurate, StyleLTC is more resource-efficient. Adding BERT embeddings or stylistic features with BERT embeddings improves results, but StyleLTC remains more efficient. When compared to large language models like LLAMA-3.1 and Mistral-7B, StyleLTC outperforms them in claim identification, even after fine-tuning. Experiment-II (Table 2 (ALL)) further confirms StyleLTC's superiority over LLAMA-3.1, achieving an F1 score of 0.86, nearly matching BERT+style's score of 0.88, while maintaining a lightweight structure.

Regarding domain extensibility, LTC models demonstrate strong adaptability across different domains. Experiment-III (Table 3) shows that training the model on one dataset and evaluating it on another results in an F1 score exceeding 85% for Datasets-ECD and GCC. However, testing on SCDC reveals a drop in Precision due to its scientific and biomedical nature. The model successfully adapts to new claim structures, as shown in Dataset-GCC, which involves Twitter data. In

Table 2: Results of experiment-I and II demonstrating performance of models across each datasets ECD, GCC, SDC and CT(for experiment-I) and when combining all of them together (ALL) for experiment-II.

	ECD			GCC			SCDC			CT			ALL		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1
BiLSTM-att	0.43	0.39	0.46	0.69	0.69	0.75	0.77	0.78	0.7	0.724	0.42	0.52	0.71	0.73	0.75
BERT-base	0.49	0.74	0.53	0.75	0.71	0.77	0.69	0.68	0.82	0.95	0.94	0.89	0.73	0.79	0.75
DistilBERT	0.73	0.64	0.68	0.79	0.91	0.85	0.61	0.95	0.73	0.85	0.78	0.81	0.81	0.84	0.82
BERT+style	0.79	0.86	0.83	0.96	0.96	0.96	0.94	0.75	0.84	0.91	0.78	0.84	0.83	0.87	0.85
LLAMA-3.1 8B (zero-shot)	0.63	0.53	0.58	0.84	0.22	0.35	0.78	0.61	0.68	0.39	0.28	0.32	0.88	0.89	0.88
LLAMA-3.1 8B (few-shot)	0.97	0.34	0.51	0.98	0.25	0.40	0.87	0.43	0.58	0.80	0.28	0.41	0.51	0.37	0.43
Mistral 7B (zero-shot)	0.39	0.98	0.56	0.38	0.84	0.53	0.44	0.82	0.57	0.41	0.8	0.54	0.9	0.44	0.59
Mistral 7B (few-shot)	0.44	0.98	0.61	0.42	0.86	0.54	0.43	0.78	0.56	0.56	0.82	0.66	0.4	0.85	0.54
Finetuned LLAMA-3.1 8B	0.88	0.54	0.67	0.82	0.39	0.53	0.89	0.62	0.73	0.85	0.65	0.74	0.61	0.82	0.70
Finetuned Mistral 7B	0.48	0.82	0.61	0.45	0.89	0.6	0.41	0.79	0.54	0.59	0.85	0.69	0.61	0.41	0.49
LTC+BERT	0.74	0.73	0.74	0.88	0.84	0.86	0.51	0.4	0.45	0.92	0.79	0.85	0.77	0.89	0.83
LTC+BERT+style	0.92	0.95	0.94	0.97	0.93	0.96	0.87	0.83	0.85	0.94	0.79	0.86	0.61	0.32	0.42
LTC+GloVe	0.69	0.62	0.65	0.79	0.53	0.64	0.53	0.23	0.32	0.82	0.58	0.68	0.73	0.29	0.41
CfC+GloVe	0.74	0.66	0.7	0.69	0.65	0.67	0.49	0.43	0.46	0.8	0.47	0.59	0.88	0.73	0.79
CfC+GloVe+style	0.92	0.87	0.9	0.98	0.99	0.98	0.79	0.76	0.78	0.91	0.82	0.86	0.85	0.86	0.85
StyleLTC (LTC+GloVe+style)	**0.92**	**0.95**	**0.94**	**0.92**	**0.98**	**0.96**	**0.86**	**0.83**	**0.84**	**0.94**	**0.81**	**0.87**	**0.88**	**0.89**	**0.88**

Table 3: Results of Experiment-III for StyleLTC when trained over a given dataset D_i (rows) and tested over other datasets D_j (columns) such that $i \neq j$.

	ECD (I)			GCC (II)			SCDC(III)			CT (IV)		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
(1)		X		0.95	0.96	0.95	0.6	0.94	0.73	0.94	0.78	0.85
(II)	0.96	0.91	0.93		X		0.52	0.93	0.67	0.86	0.79	0.82
(III)	0.95	0.65	0.77	0.98	0.97	0.97		X		0.94	0.79	0.86
(IV)	0.96	0.64	0.77	0.99	0.96	0.98	0.85	0.78	0.81		X	

Table 4: Performance of models on skewed dataset

Model	P	R	F1
BiLSTM-att	0	0	0
BERT-base	0	0	0
DistilBERT	0	0	0
LTC+GloVe	0	0	0
CfC+GloVe	0	0	0
CfC+GloVe+style	0.88	0.65	0.75
BERT+style	0.231	0.017	0.031
LTC+BERT	0.57	0.0074	0.015
LTC+BERT+style	0.93	0.723	0.813
StyleLTC (LTC+GloVe+style)	0.93	0.74	0.82

terms of performance on highly skewed datasets, the StyleLTC model outperforms traditional models like BiLSTM and BERT when applied to a modified Dataset-SCDC (Table 4). In this skewed dataset, where over 80% of sentences are labeled as not-claim, StyleLTC achieves an F1 score of 82%, while traditional models struggle, with BERT+style scoring only 0.31%. This highlights the effectiveness of the LTC model in handling skewed datasets and underscores the importance of incorporating stylistic vectors in claim detection.

Resource Efficiency: LTC models are highly resource-efficient in terms of FLOP counts (see Table 5 in Appendix D), making them suitable for CPU memory operations, unlike transformer-based models like BERT, which require substantial GPU resources. For instance, the BERT-base model demands around 20.3 GB of GPU memory, while LTC and StyleLTC models use less than 1 MB (see Figure 4 in Appendix D), highlighting their minimal memory requirements. When compared to large-scale models like LLAMA-3.1 (8B), the LTC-based text identification model, which uses only 100K parameters and 70 neurons in AutoNCP wiring, outperforms LLAMA-3.1 during inference. Training large language models is resource- and time-intensive, while the LTC model trains much faster—StyleLTC, for example, trains for 50 epochs on domain-specific datasets in just 1 hour. Additionally, as shown in Figure 5 Appendix D, the training time per epoch for Dataset-CT is minimal, and prediction times per sample are only in the milliseconds range.

4 Conclusion

The paper introduced style-infused liquid neural networks for claim identification task, showing their ability to capture the intricate temporal dynamics present in the sequential nature of language. The StyleLTC model has consistently outperformed baseline models, particularly in noisy, cross-domain environments, providing significant advantages in both accuracy and resource efficiency. Evaluation across diverse domains highlights that StyleLTC excels in specialized settings, solidifying its position as an alternative to conventional models in claim identification tasks.

5 Limitations

While a standout feature of liquid neural networks is their proficiency in handling time-series data, this study limited the time dimension to one, relying solely on text as sequential data. Despite this constraint, our style-aware LTC model has demonstrated a notable capacity to exceed state-of-the-art baselines and, in certain cases, even outperform large language models in resource-constrained environments. This suggests that even within a restricted temporal framework, the model can effectively leverage linguistic nuances to achieve impressive results. Our experiments have consistently demonstrated that incorporating textual embeddings with stylistic features leads to enhanced performance. We have explored both pre-trained BERT and GloVe embeddings in our work. Looking ahead, we aim to investigate the use of more contextually rich embeddings. Detailed specifications and parameters of our model and experiments can be found in Section 2 and 3.

Future Work: The potential applications of Liquid Neural Network-based models extend beyond claim identification; they offer a chance to enhance the explainability of the model's decision-making processes, particularly concerning the verifiability of claims. Our future research will prioritize this critical aspect, seeking to unpack the mechanisms behind the model's predictions. Moreover, our experiments indicate that integrating text embeddings with grammatical vectors and stylistic scores results in a significant improvement in F1 scores for StyleLTC. As we move forward, we aim to investigate which specific stylistic features contribute most substantially to performance enhancements.

By further refining our model and examining its versatility across a broader range of contexts, we aspire to contribute to the ongoing development of Liquid Neural Networks and their implementation in natural language processing, ultimately advancing the field's understanding of how sophisticated temporal analysis can enhance language comprehension and processing capabilities.

References

Titipat Achakulvisut, Chandra Bhagavatula, Daniel Acuna, and Konrad Kording. 2019. Claim extraction in biomedical publications using deep discourse model and transfer learning. *arXiv preprint arXiv:1907.00962*.

Firoj Alam, Shaden Shaar, Fahim Dalvi, Hassan Sajjad, Alex Nikolov, Hamdy Mubarak, Giovanni Da San Martino, Ahmed Abdelali, Nadir Durrani, Kareem Darwish, et al. 2020. Fighting the covid-19 infodemic: Modeling the perspective of journalists, fact-checkers, social media platforms, policy makers, and the society. *arXiv preprint arXiv:2005.00033*.

Fatma Arslan, Naeemul Hassan, Chengkai Li, and Mark Tremayne. 2020. A benchmark dataset of check-worthy factual claims. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 14, pages 821–829.

Pepa Atanasova, Alberto Barron-Cedeno, Tamer Elsayed, Reem Suwaileh, Wajdi Zaghouani, Spas Kyuchukov, Giovanni Da San Martino, and Preslav Nakov. 2018. Overview of the clef-2018 checkthat! lab on automatic identification and verification of political claims. task 1: Check-worthiness. *Preprint*, arXiv:1808.05542.

Alberto Barrón-Cedeño, Firoj Alam, Tommaso Caselli, Giovanni Da San Martino, Tamer Elsayed, Andrea Galassi, Fatima Haouari, Federico Ruggeri, Julia Maria Struß, Rabindra Nath Nandi, et al. 2023. The clef-2023 checkthat! lab: Checkworthiness, subjectivity, political bias, factuality, and authority. In *European conference on information retrieval*, pages 506–517. Springer.

Alberto Barrón-Cedeño, Firoj Alam, Tanmoy Chakraborty, Tamer Elsayed, Preslav Nakov, Piotr Przybyła, Julia Maria Struß, Fatima Haouari, Maram Hasanain, Federico Ruggeri, Xingyi Song, and Reem Suwaileh. 2024. The clef-2024 checkthat! lab: Check-worthiness, subjectivity, persuasion, roles, authorities, and adversarial robustness. In *Advances in Information Retrieval*, pages 449–458, Cham. Springer Nature Switzerland.

Alberto Barron-Cedeno, Tamer Elsayed, Preslav Nakov, Giovanni Da San Martino, Maram Hasanain, Reem Suwaileh, Fatima Haouari, Nikolay Babulkov, Bayan Hamdan, Alex Nikolov, Shaden Shaar, and Zien Sheikh Ali. 2020. Overview of checkthat! 2020: Automatic identification and verification of claims in social media. *Preprint*, arXiv:2007.07997.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms. In *Advances in Neural Information Processing Systems*, volume 36, pages 10088–10115. Curran Associates, Inc.

Tommaso Fornaciari, Dirk Hovy, Elin Naurin, Julia Runeson, Robert Thomson, and Pankaj Adhikari. 2021. "we will reduce taxes" - identifying election pledges with language models. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3406–3419, Online. Association for Computational Linguistics.

Ken-ichi Funahashi and Yuichi Nakamura. 1993. Approximation of dynamical systems by continuous time recurrent neural networks. *Neural networks*, 6(6):801–806.

Shreya Gupta, Parantak Singh, Megha Sundriyal, Md Shad Akhtar, and Tanmoy Chakraborty. 2021. Lesa: Linguistic encapsulation and semantic amalgamation based generalised claim detection from online content. *arXiv preprint arXiv:2101.11891*.

Casper Hansen, Christian Hansen, Stephen Alstrup, Jakob Grue Simonsen, and Christina Lioma. 2019. Neural check-worthiness ranking with weak supervision: Finding sentences for fact-checking. In *Companion proceedings of the 2019 world wide web conference*, pages 994–1000.

Ramin Hasani, Mathias Lechner, Alexander Amini, Lucas Liebenwein, Aaron Ray, Max Tschaikowski, Gerald Teschl, and Daniela Rus. 2022. Closed-form continuous-time neural networks. *Nature Machine Intelligence*, 4(11):992–1003.

Ramin Hasani, Mathias Lechner, Alexander Amini, Daniela Rus, and Radu Grosu. 2021. Liquid time-constant networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 7657–7666.

Naeemul Hassan, Fatma Arslan, Chengkai Li, and Mark Tremayne. 2017. Toward automated fact-checking: Detecting check-worthy factual claims by claimbuster. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1803–1812.

Zhaojing Huang, Luis Fernando Herbozo Contreras, Wing Hang Leung, Leping Yu, Nhan Duy Truong, Armin Nikpour, and Omid Kavehei. 2024. Efficient edge-ai models for robust ecg abnormality detection on resource-constrained hardware. *Journal of Cardiovascular Translational Research*, pages 1–14.

Israa Jaradat, Pepa Gencheva, Alberto Barrón-Cedeño, Lluís Màrquez, and Preslav Nakov. 2018. ClaimRank: Detecting check-worthy claims in Arabic and English. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, pages 26–30, New Orleans, Louisiana. Association for Computational Linguistics.

Yavuz Selim Kartal and Mucahid Kutlu. 2022. Rethink before you share: A comprehensive study on prioritizing check-worthy claims. *IEEE transactions on computational social systems*, 10(1):362–375.

Lev Konstantinovskiy, Oliver Price, Mevan Babakar, and Arkaitz Zubiaga. 2021. Toward automated factchecking: Developing an annotation schema and benchmark for consistent automated claim detection. *Digital threats: research and practice*, 2(2):1–16.

Mathias Lechner, Ramin Hasani, Alexander Amini, Thomas A Henzinger, Daniela Rus, and Radu Grosu. 2020. Neural circuit policies enabling auditable autonomy. *Nature Machine Intelligence*, 2(10):642–652.

Preslav Nakov, Alberto Barrón-Cedeño, Giovanni Da San Martino, Firoj Alam, Rubén Míguez, Tommaso Caselli, Mucahid Kutlu, Wajdi Zaghouani, Chengkai Li, Shaden Shaar, et al. 2022. Overview of the clef-2022 checkthat! lab task 1 on identifying relevant claims in tweets. In *2022 Conference and Labs of the Evaluation Forum, CLEF 2022*, pages 368–392. CEUR Workshop Proceedings (CEUR-WS. org).

Jingwei Ni, Minjing Shi, Dominik Stammbach, Mrinmaya Sachan, Elliott Ash, and Markus Leippold. 2024. Afacta: Assisting the annotation of factual claim detection with reliable llm annotators. *arXiv preprint arXiv:2402.11073*.

Rrubaa Panchendrarajan and Arkaitz Zubiaga. 2024. Claim detection for automated fact-checking: A survey on monolingual, multilingual and cross-lingual research. *Natural Language Processing Journal*, 7:100066.

James W Pennebaker, Ryan L Boyd, Kayla Jordan, and Kate Blackburn. 2015. The development and psychometric properties of liwc2015. Technical report.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Peng Qi, Zehong Yan, Wynne Hsu, and Mong Li Lee. 2024. Sniffer: Multimodal large language model for explainable out-of-context misinformation detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13052–13062.

Dorian Quelle and Alexandre Bovet. 2024. The perils and promises of fact-checking with large language models. *Frontiers in Artificial Intelligence*, 7:1341697.

Michael Schlichtkrull, Zhijiang Guo, and Andreas Vlachos. 2024. Averitec: A dataset for real-world claim verification with evidence from the web. *Advances in Neural Information Processing Systems*, 36.

Manjira Sinha, Nilesh Agarwal, and Tirthankar Dasgupta. 2020. Relation aware attention model for uncertainty detection in text. In *Proceedings of the ACM/IEEE Joint Conference on Digital Libraries in 2020*, pages 437–440.

Dominik Stammbach, Nicolas Webersinke, Julia Anna Bingler, Mathias Kraus, and Markus Leippold. 2023. Environmental claim detection. *Preprint*, arXiv:2209.00507.

Shivashankar Subramanian, Trevor Cohn, and Timothy Baldwin. 2019. Deep ordinal regression for pledge specificity prediction. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*

(EMNLP-IJCNLP), pages 1729–1740, Hong Kong, China. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Vinicius Woloszyn, Joseph Kobti, and Vera Schmitt. 2021. Towards automatic green claim detection. In *Proceedings of the 13th Annual Meeting of the Forum for Information Retrieval Evaluation*, pages 28–34.

Qing-Song Xu and Yi-Zeng Liang. 2001. Monte carlo cross validation. *Chemometrics and Intelligent Laboratory Systems*, 56(1):1–11.

Zhenxun Zhuang, Mingrui Liu, Ashok Cutkosky, and Francesco Orabona. 2022. Understanding adamw through proximal methods and scale-freeness. *Transactions on Machine Learning Research*.

A Data Descriptions

A.1 Dataset-CT: Checkworthy English Dataset

This dataset is obtained from Check-Worthiness Estimation task of *CheckThat! Lab at CLEF 2024* (Barrón-Cedeño et al., 2024). The aim of this task is to determine whether a claim in a tweet or transcriptions is worth fact-checking. The binary identification task is available in Arabic, English, and Spanish. We have taken the expertly annotated english dataset which are labeled as "yes" and "no" to detect "claim" and "not-claim" respectively. The training dataset consists 22501 sentences among which 5413 are "claim" and 17088 are "not-claim". The test dataset consists 1350 texts among which 346 are claims and 1004 are not-claims.

A.2 Dataset-ECD: Environmental Claim Detection Dataset

We have collected the environmental claim dataset available from (Stammbach et al., 2023). The dataset contains environmental claims made by listed companies. The authors have collected text from sustainability reports, earning calls, and annual reports of listed companies and annotated 3000 sentences. After discarding tied annotations, the final dataset contains 2647 examples. There are 665 claim statements and 1982 not claim statements.

A.3 Dataset-GCC: Green-Claims Corpus

We choose the Automatic Green Claims Detection corpus consisting of 773 tweets from domains such as cosmetics and electronics (Woloszyn et al., 2021). All the tweets are classified into two classes "green-claim" and "not green-claim". For Binary Classification, there are 506 "not green-claim" and 267 "green-claims".

A.4 Dataset-SCDC: Scientific Claim Detection Corpus

To test the generalizability of the proposed model in biomedical domain, we took the dataset mentioned in (Achakulvisut et al., 2019). The dataset includes text extracts from expertly annotated 11519 claims in biomedical paper abstracts. Here the dataset is labeled into six classes: "False", "barely-true", "half-true", "pants-fire", "barely-false", and "True".

B Determining Model Parameters

As discussed in Section 2, we determine the optimal number of sensory neurons by analyzing the F1 scores for Political Bias detection in News Articles and News Media (available in English), as specified in Subtask 3A of the CheckThat! Lab at CLEF 2023 (Barrón-Cedeño et al., 2023). The corresponding plot is presented in Figure 3. To identify the optimal configuration, we systematically increase the number of sensory neurons in the model and evaluate performance on the specified task. As observed from the graph, the highest F1 score is achieved when the number of sensory neurons in the LTC model is set to 70. Consequently, we adopt this configuration for our StyleLTC model.

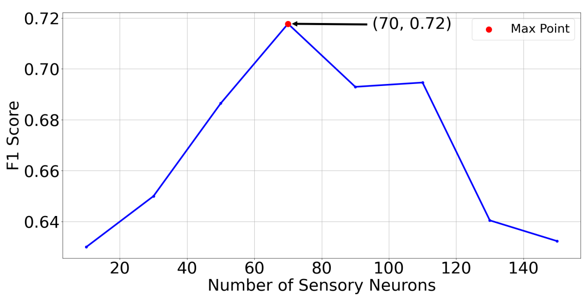

Figure 3: No. of sensory neurons of StyleLTC Model vs corresponding F1 Scores of the Model

C Closed-Form Continuous-Time Neural Networks

Continuous neural network architectures based on ordinary differential equations serve as effective models for capturing complex data dynamics (Hasani et al., 2022). They transform the depth dimension of static neural networks and the time dimension of recurrent networks into a continuous vector field, facilitating parameter sharing, adaptive computations, and efficient function approximation for non-uniformly sampled data. The closed-form solution for neuron-synapse interactions in continuous-time neural networks provides significant efficiency enhancements, allowing training and inference to be between one and five orders of magnitude faster than models reliant on numerical differential equation solvers. Additionally, the Closed-form control (CfC) derived from liquid time-constant dynamics exhibits notable scalability and performance in time-series modeling, making it applicable across various domains. The CfC model consists of an input perception module, an LTC module, and outputs. A distinguishing feature of CfC neural networks is their independence from numerical ordinary differential equation (ODE) solvers for generating temporal rollouts, thereby achieving the flexible, causal, and continuous-time characteristics of ODE-based networks while enhancing efficiency (Huang et al., 2024). In this study, we employed the CFC network for identifying claim and non-claim sentences.

C.1 Training and Validation of CFC Network

The CFC network was trained using the training dataset from each discussed dataset: ECD, GCC, SCDC, and CT. To ensure reproducibility, a random seed of 42 was set on the CPU. The PyTorch Lightning Trainer was initialized with an early stopping callback to monitor validation loss, halting training after 20 epochs without improvement. The AdamW optimization algorithm was employed to minimize the mean squared error loss function, with training configured for a maximum of 100 epochs and gradient clipping set at 1 for stability. After each epoch, validation metrics such as loss, precision, recall, and F1 score were recorded to track performance and guide early stopping. In the CFC+GloVe configuration, we provided 300-dimensional GloVe embeddings of the text as input, while in the CFC+GloVe+style configuration, we concatenated the 300-dimensional GloVe embeddings with 76-dimensional stylistic features for input.

D Parameter efficiency of StyleLTC

In Figure 4, we present plots depicting epoch vs. accuracy, epoch vs. validation loss, and epoch vs. memory usage for StyleLTC, LTC+GloVe, and the pre-trained BERT model. Subfigures (a), (b), (c), and (d) correspond to the results obtained on Datasets ECD, GCC, SCDC, and CT, respectively. Based on these plots, we derive several key observations. Liquid Neural Network-based models exhibit exceptional resource efficiency, making them highly suitable for CPU-based memory operations. This stands in contrast to transformer-based models such as BERT, which require significant GPU resources. As shown in Figure 4, the BERT-base model consumes approximately 20.3 GB of GPU memory, whereas the LTC and StyleLTC models require less than 1 MB, underscoring their minimal memory footprint relative to transformer-based architectures. Furthermore, we compared LTC models with large-scale models such as LLAMA-3.1 (8 billion parameters) and observed that LLAMA-3.1 demonstrated inferior inference performance compared to our LTC-based text identification model. The LTC-based model operates with only 100K parameters and 70 neurons in the AutoNCP wiring, highlighting its computational efficiency. Training large language models is both computationally and time-intensive, whereas the LTC model offers a significantly faster training process. For example, the StyleLTC model requires only one hour to train for 50 epochs on domain-specific datasets. Additionally, as illustrated in Figure 5, the training time per epoch for Dataset-CT is minimal, and the prediction time per sample is measured in milliseconds, further emphasizing the efficiency of the LTC-based approach.

Table 5: FLOPs Parameter count for LTC model trained on Dataset-CT

Name	Shape
rnn cell.gleak	(70,)
rnn cell.vleak	(70,)
rnn cell.cm	(70,)
rnn cell.sigma	(70,70)
rnn cell.mu	(70,70)
rnn cell.w	(70,70)
rnn cell.erev	(70,70)
rnn cell.sensory sigma	(376,70)
rnn cell.sensory mu	(376,70)
rnn cell.sensory w	(376,70)
rnn cell.sensory erev	(376,70)
rnn cell.sparsity mask	(376,70)
rnn cell.sensory sparsity mask	(376,70)
rnn cell.input w	(376,)
rnn cell.input b	(376,)
rnn cell.output w	(1,)
rnn cell.output b	(1,)

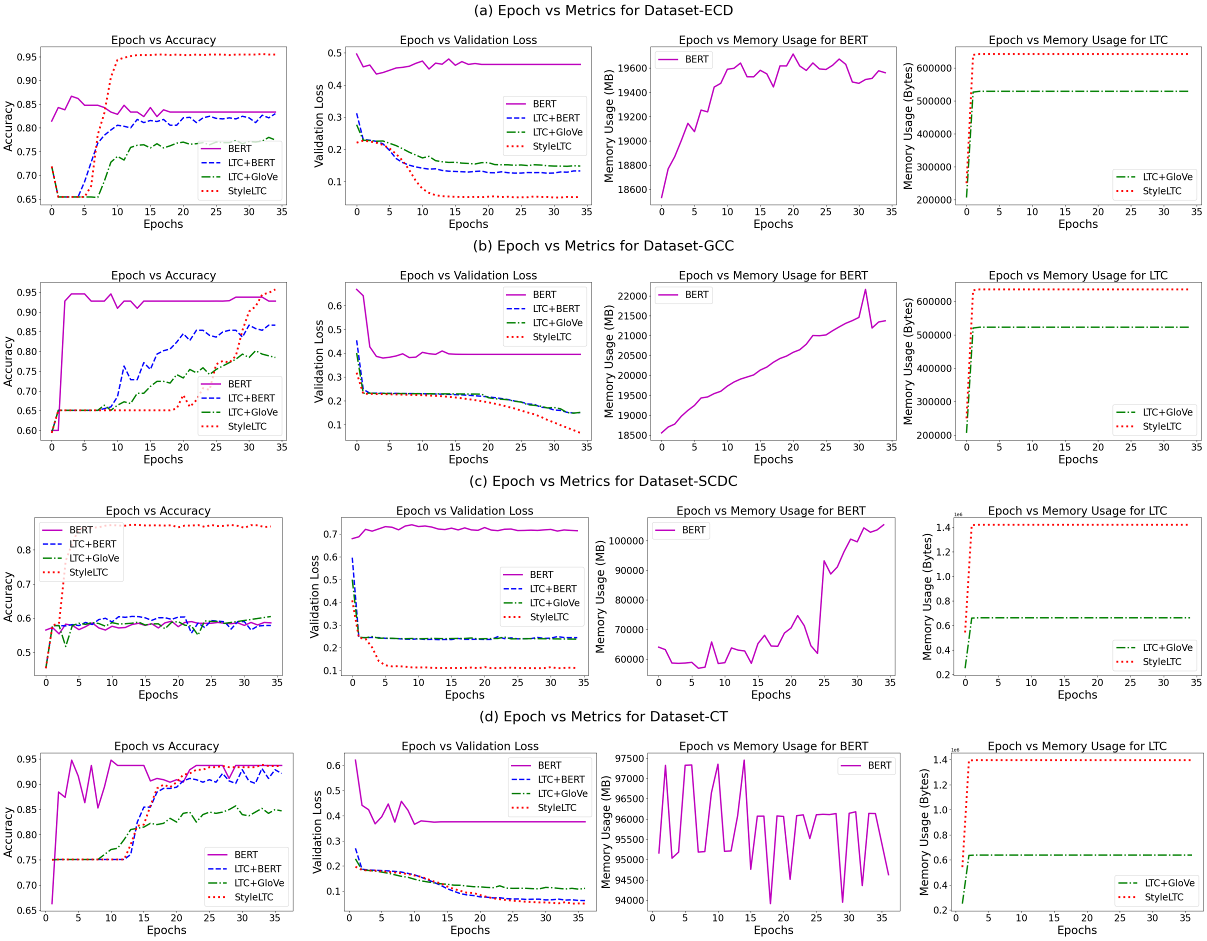

Figure 4: Epoch Vs Metrics and Memory Usage for Different Datasets

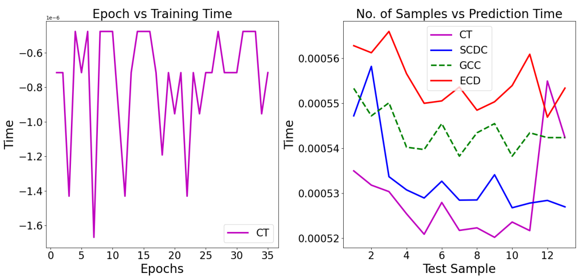

Figure 5: Training time per epoch and prediction time per sample for StyleLTC across various datasets

E Fine-Tuning LLama 3.1 and Mistral 7B for Claim Detection

We fine-tune LLama 3.1 8B and Mistral 7B using transfer learning, incorporating quantization and LoRA adapters (Dettmers et al., 2023) to enhance efficiency and adaptability for claim detection. The process begins with loading and processing one of the datasets— ECD, GCC, SCDC, or CT—where each sample is augmented with a task-specific prompt. To improve computational efficiency, we apply quantization, reducing the model's precision from 32-bit to 4-bit. This enables faster computation and lower memory usage with minimal impact on accuracy. Additionally, LoRA adapters are integrated into specific model layers, enabling task-specific fine-tuning while preserving the pre-trained weights. The training setup consists of a batch size of 8, the AdamW optimizer (Zhuang et al., 2022), a learning rate of $2 * e^{-4}$ with a cosine scheduler, and 50 training epochs with a maximum of 500 steps. Gradient accumulation is employed to reduce memory consumption, and learning rate scheduling ensures stable convergence. During training, the model processes batches of data, making predictions that are compared with ground truth labels to compute loss, which is minimized via backpropagation. Logging occurs at regular intervals to monitor progress, loss, and resource usage, while evaluation metrics are periodically computed on validation data to assess generalization and detect potential overfitting.