Supplementary data

Appendix 1: Summary of reviewed papers

S/N	Articles	Country, HDI Rank	Journal	Model used
1	Abbas et al. (2020)	Lao People's Democratic Republic	Journal of Hydrology	LSTM, HSPF, HRU-based LSTM
2	Afan et al. (2022)	Iraq	Natural Hazards	Linear and stratified DL models (unclear)
3	Bui et al. (2020)	Vietnam	Journal of Hydroinformatics	Grasshopper, Grey Wolf and Social Spider Optimizations, DNN
4	Cai and Yu (2022)	China	Urban Climate	Hybrid RNN (CNN, LSTM, Bi-LSTM + ARIMAX)
5	Chen et al. (2015)	China	Neural processing letter	RNN with Elman network
6	Chen et al. (2021)	Iran	Geocarto International	CNN, CNN-GWO, CNN-ICA
7	Chen et al. (2021)	China	Computer Networks	CNN with different batch normalizations
8	Chhetri et al. (2020)	Bhutan	Remote Sensing (MDPI)	LR, MLP, CNN, LSTM, GRU and BiLSTM
9	Cui et al. (2021)	China	Hydrology Research	XAJ, LSTM, XAJ-LSTM
10	Cui et al. (2022)	China	Journal of Hydrology	XAJ, LSTM, LSTM-RED, LSTM-EDE
11	Dey et al. (2021)	India	Water Resources Management	BiLSTM - LSTM ensemble, BiLSTM, BHLSTM
12	Endalie et (2021)	Ethiopia	Water Supply	LSTM, MLP, KNN, SWM, DT
13	Ha et al. (2021)	China	Scientific Reports	Stacked LSTM, Cov LSTM encoder-decoder LSTM, Conv LSTM encoder-decoder GRU
14	Habumugisha et al. (2022)	China	Sustainability	CNN, DNN, LSTM and RNN
15	Hien Than et al. (2021)	Vietnam	Journal of Hydrology	LSTM, ARIMA, NAR-MA, LSTM-MA
16	Hu et al. (2018)	China	Water	ANN, LSTM
17	Jiang et al. (2022)	China	Environmental Modelling and Software	LSTM, CNN, RF, and MLP
18	Kang et al. (2020)	China	Atmosphere	ARMA, MARS, BPNN, SVM, GA, LSTM
19	Kanth et al. (2022)	India	Stochastic Environmental Research and Risk Assessment	ANN, BERT, Bi-LSTM, CNN

20	Kardhana et al. (2022)	Indonesia	Water	ANN, Simple RNN, LSTM-RNN
21	Kaur et al. (2021)	India	Natural Hazards	PCA, ANN
22	Khosravi et al. (2020)	Iran	Journal of Hydrology	CNN
23	Kumar et al (2022)	India	Stochastic Environmental Research and Risk Assessment	Vanilla and Stacked LSTMs
24	Kumar et al. (2004)	India	Water Resources Management	FFN, RNN
25	Kumar et al. (2019)	India	Hydrological Sciences Journal	RNN and LSTM
26	Le et al. (2019)	Vietnam	Water	LSTM
27	Liu et al. (2020)	China	Water	RNN, XAJ, LSTM, LSTM-KNN
28	Liu et al. (2022)	China	Hydrology and Eath Sciences	LSTM, Meteo-Hydro-LSTM, ESP-Hydro, Meteo-Hydro
29	Loganathan and Mahindrakar (2021)	India	Journal of Water & Climate	NNET and ML models: EXGBDT, DT, KNN, PLS, GLM and PCR
30	Lv et al. (2020)	China	Advances in Water Resources	LSTMC,BPNNC, LRC
31	Maddu et al. (2021)	India	Journal of Water & Climate	ANN, LSTM, LSTM-BiLSTM
32	Noor et al. (2022)	Bangladesh	Water	ANN, LSTM, TALSTM ,SALSTM, STALSTM,
33	Pereira Filho & Dos Santos (2006)	Brazil	Journal of Hydrology	ANN, ARIMA
34	Sankaranarayanan et al. (2020)	India	Journal of Water and Climate Change	Deep neural network, KNN, SVM, and Naïve Bayes
35	Song et al. (2020)	China	Water (MDPI)	LSTM, LSTM- flash flood (LSTM-FF)
36	Sun et al. (2022)	China	Environmental Research Communications	BP, LSTM, ELM, PSO-LSTM, HHO-LSTM, IHHO-LSTM, VMD-IHHO-LSTM, OVMD-IHHO-LSTM
37	Tikhamarine et al. (2020)	Egypt	Journal of Hydrology	ANN-GWO, ANN, SVR-GWO, SVR, MLR_GWO
38	Van et al. (2020)	Vietnam	Journal of Hydroinformatics	CNN, LSTM
39	Wang et al. (2022)	China	Water Supply	GRU, RF, SVR
40	Xu et al. (2021)	China	Hydrology Research	TCN, TCN (NDVI), LSTM, EIESM, ANN
41	Xu et al. (2022)	China	Journal of Hydrology	LSTM, PSO-LSTM, ANN, PSO-ANN
42	Xu et al. (2022)	China	Frontiers in Earth Science	CNN-LSTM, CNN-GRU, SWAT,
43	Xu et al. (2022)	China	Water Resources Management	CNN-GRU,

44	Yeditha et al. (2021)	India	Journal of Hydroinformatics	ANN, ELM and LSTM
45	Zhang et al. (2021)	China	Water	MSBP and Random Forest
46	Zhang et al. (2022)	China	Acta Geotechnica	ANN, GRU, RF, MARS
47	Zhang et al. (2022)	China	IEEE Access	LSTM, GRU, TCN, ConvLSTM,TCN-Attn-RNN, RNN-Attn-TCN
48	Zhou et al. (2020)	China	Water	NARX, BPNN
49	Zhu et al. (2020)	China	Sensors	Cascade-parallel LSTM-CRF, MLP, Logistic regression, decision tree

LSTM: long short term memory, HSPF: hydrological simulated program-FORTRAN, DNN: deep neural network, CNN: convolutional neural network, GWO: grey wolf optimizer, ICA – imperialist competitive algorithm, LR: linear regression, MLP: multi-layer perceptron, GRU: gated recurrent unit, BiLSTM: bidirectional LSTM, KNN: k-nearest neighbors, SVM: support vector machine, DT: decision tree, BERT: bidirectional encoder representations from transformers, PCA/PCR: principal component analysis/regression, FFN: feed forward network, NNET: neural network, EXGBDT: extreme gradient boosting decision tree, PLS: partial least-squared regression, GLM: generalized linear model, ARIMA: autoregressive integrated moving average, NAR-MA: non-linear autoregressive-moving average neural network, LSTM-MA: moving average long short term memory, ELM: extreme learning machine. TS: Threat Scores, DBN: Deep Belief Network, CDBN: Convolutional DB Network, CRPS: Continuous Ranked Probability Score, XAJ: Xinanjiang model, LSTMC: LSTM cyclic, LSTMC: LSTM cyclic, LRC: Linear regression cyclic, BPNNC: Back propagation neural network cyclic, STALSTM: spatio-temporal attention LSTM, ANN: Artificial Neural Network, TCN: Temporal Convolutional Network, EIESM: Excess Infiltration and Excess Storage Model , MSBP Multi-Step Back Propagation, RF: random forest, MARS: Multivariate adaptive regression splines, MLP: Multilayer perceptron, LSTM-RED: Recursive Encoder-Decoder LSTM, LSTM-EDE: Exogenous Encoder-Decoder LSTM, FFN: Feed Forward Network, BHLSTM: Bilstm with highway network, CNN-ICA: CNN and imperialist competitve algorithm, SSO: Social Spider Optimizations, NARX: Non-linear auto-regressive with exogenous input neural network, BPNN: Back propagation neural network, MLPR : Multi perceptron regression, Cascade-parallel LSTM-CRF: Cascade-parallel LSTM Conditional Random Field

MBE: Mean deviation, CE: NSE : Nash Sutcliffe Efficiency, VE: Volume Error, RE: Relative Error, CC: Coefficient of Certainty, WI: Willmott's Index of Agreement, LMI: Legates-McCabe's Index, MAPE: Mean Percentage Error, b: bias factor, AbsErr: Absolute Error, MRE: Mean relative error, MAE: Mean Absolute Error, CRF: Conditional Random Field, AUC: Area under curve, TA: Total accuracy, PPR: Positive predictive rate, NPR: negative predictive rate, PPTS: peak percent threshold statistics, SDL: Stratified deep learning models, SSO: Social Spider Optimized DNN

Appendix 2: Trend Analysis

Hypothesis ¶

Ho = Null hypothesis: There is no trend in published articles over the last 3 decades Ha = Alternative hypothesis: There is a trend in published articles over the last 3 decades

p = 0.05

Articles in developing countries

In [10]:
```
import pymannkendall as pymk

art_dev = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0
pymk.original_test(art_dev)
```

Out[10]:
```
Mann_Kendall_Test(trend='increasing', h=True, p=0.0004528936968959396, z=3.
507173533407478, Tau=0.35172413793103446, s=153.0, var_s=1878.333333333333
3, slope=0.0, intercept=0.0)
```

In [14]:
```
trend, h, p, z, Tau, s, var_s, slope, intercept = pymk.original_test(art_
trend, h, p, z, Tau, s, var_s, slope, intercept
```

Out[14]:
```
('increasing',
 True,
 0.0004528936968959396,
 3.507173533407478,
 0.35172413793103446,
 153.0,
 1878.3333333333333,
 0.0,
 0.0)
```

In [18]:
```
import matplotlib.pyplot as plt
import pandas as pd
df_dev = pd.read_csv("articles in developing countries.csv", index_col =
df_dev.head()
```

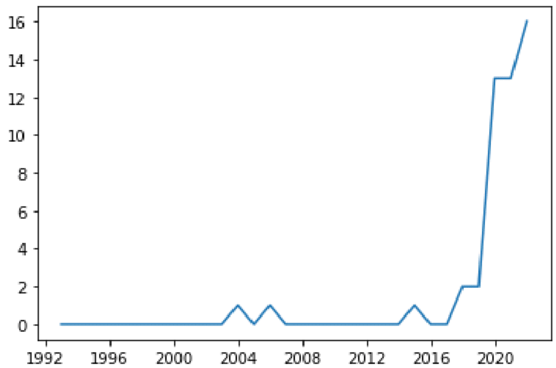
Out[18]:

	Reviewed articles
Date	
1993-01-01	0
1994-01-01	0
1995-01-01	0
1996-01-01	0
1997-01-01	0

In [19]:

```
plt.plot(df_dev)
```

Out[19]: [<matplotlib.lines.Line2D at 0x256703c2940>]



Global data

In [23]:

```
df_glob = pd.read_csv("published articles globally.csv", index_col = 'Da
df_glob.head()
```

Out[23]:

Date	Articles published globally
1993-01-01	2
1994-01-01	5
1995-01-01	0
1996-01-01	3
1997-01-01	10

In [24]:

```
pymk.original_test(df_glob)
```

Out[24]:

```
Mann_Kendall_Test(trend='increasing', h=True, p=1.715738662255717e-12, z=7.
055821730567926, Tau=0.9103448275862069, s=396.0, var_s=3134.0, slope=2.529
4117647058822, intercept=-19.67647058823529)
```

In [26]:

```
trend, h, p, z, Tau, s, var_s, slope, intercept = pymk.original_test(df_g
trend, h, p, z, Tau, s, var_s, slope, intercept
```

Out[26]:

```
('increasing',
 True,
 1.715738662255717e-12,
 7.055821730567926,
 0.9103448275862069,
 396.0,
 3134.0,
 2.5294117647058822,
 -19.67647058823529)
```

In [30]:

```
print(p)
```

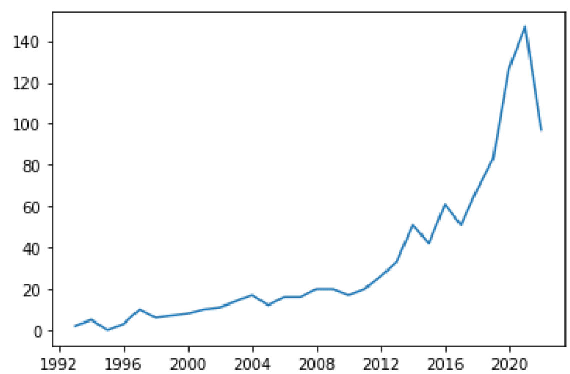
```
1.715738662255717e-12
```

In [27]:

```
plt.plot(df_glob)
```

Out[27]:

```
[<matplotlib.lines.Line2D at 0x25674249100>]
```



In [ ]:

```

```