CS 5970

ANNE Project

Comparison of Time Series Forecasting using Feed Forward Neural Network, Recurrent Neural Network and Hybrid Method

Egawati Panjei - 112866032

Outline

- **Introduction**
- Project Objectives
- Implementation
- Experimental Design
- Result
- Conclusion
- Future Work
- References

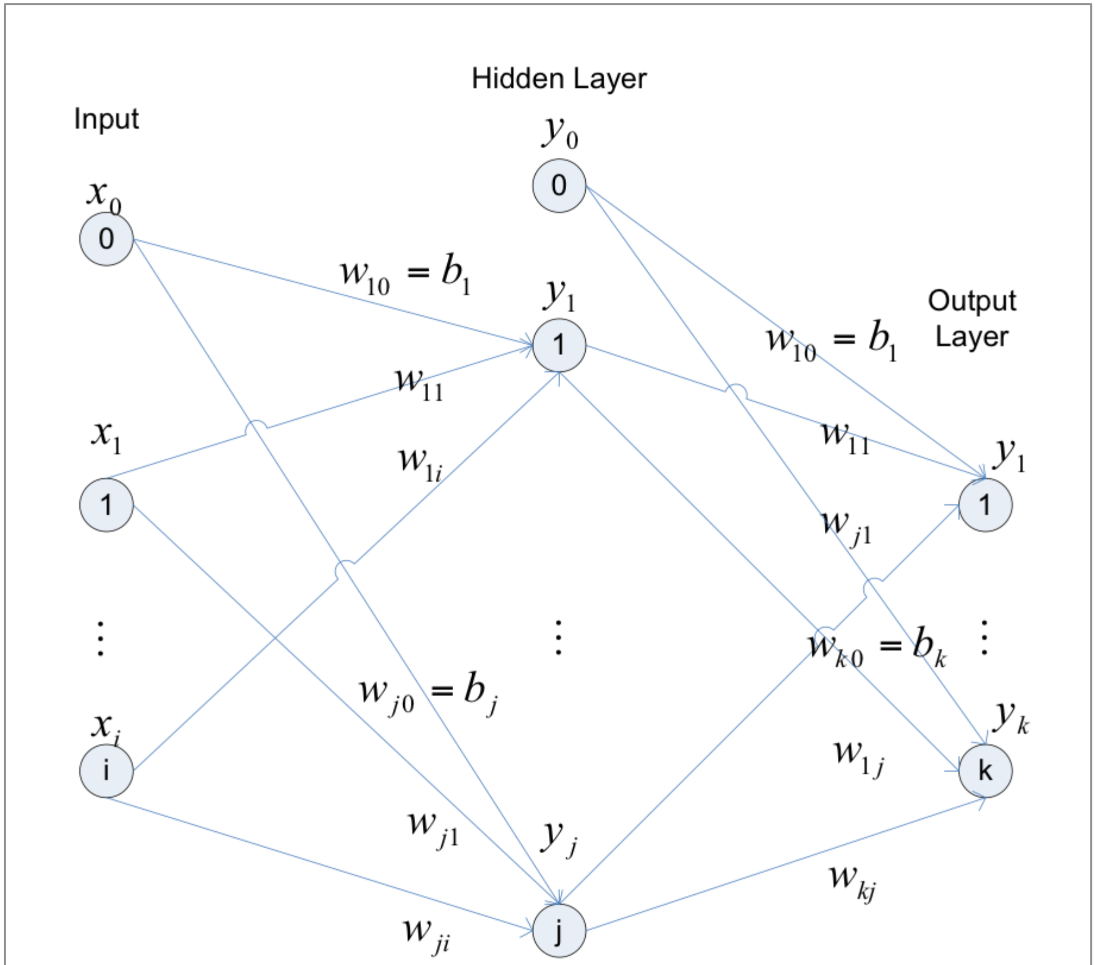
Time Series Forecasting

- Time series forecasting have been applied in many areas like:
  - environment : daily rain fall, weekly temperature
  - industry : hourly electricity consumption, quarterly product manufactured
  - economy : monthly profit, monthly sales volume, future trading
- Time series forecasting type:
  - Single step time series
  - Multi step time series
  - Step-wise time series

Time Series Forecasting

- Methods used for time series forecasting
  - Linear Method : Linear regression (LR), exponential smoothing (ES), autoregressive integrated moving average (ARIMA) to predict linear time series.
  - Non Linear Method : Neural Network
  - Hybrid Method : combination of linear method and non linear method
- This project deals with comparison of Non Linear Method and Hybrid Method

Feed Forward Neural Network



Input for every hidden layer unit v_j:

$$net_j = \sum_{i=0} w_{ji} x_i$$

The output of each neuron:

$$y_j = \phi_j(net_j)$$

$$\phi_j(net_j) = \frac{1}{1 + \exp(-net_j)}$$

For every neuron in output layer:

$$net_k = \sum_{i=0} w_{kj} y_j$$

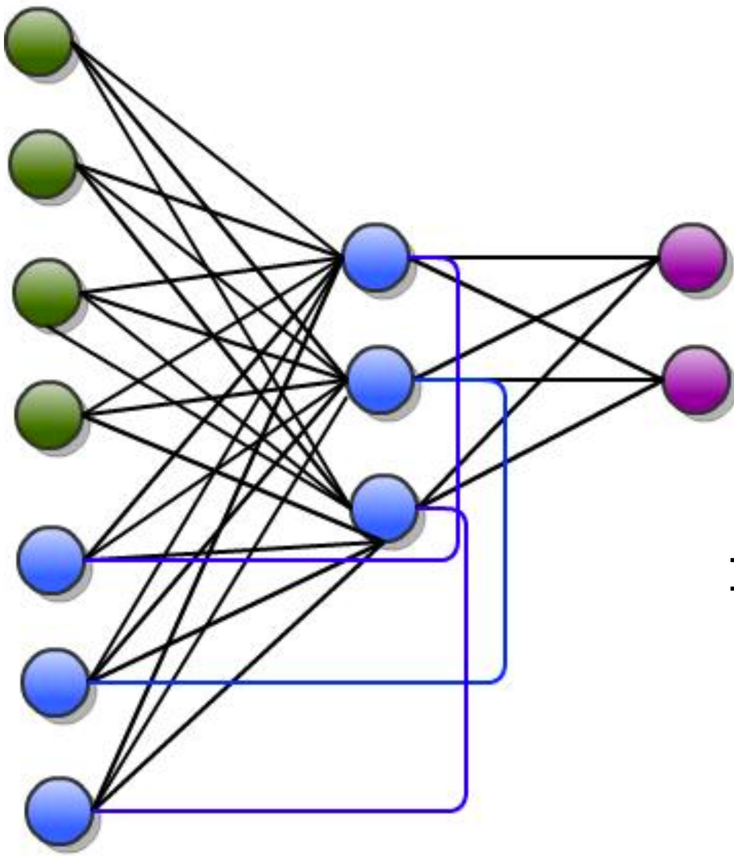
$$o_k = \phi_k(net_k)$$

Feed Forward Neural Network

- Standard backpropagation algorithm for FFNN:

1. **Compute error information terms of output layer** $\boldsymbol{\delta_k = (t_k - o_k) f'(net_k)}$
2. **Calculate delta weights between hidden and output layer** $\boldsymbol{\Delta w_{jk} = \alpha \delta_k y_j}$
3. **Compute error terms of hidden layer** $\boldsymbol{\delta_j = \left(\sum_k^K \delta_k w_{kj}\right) f'(net_j)}$
4. **Calculate delta weights between hidden and output layer** $\boldsymbol{\Delta w_{ij} = \eta \delta_j x_i}$
5. **Update weights** $\boldsymbol{w_{new} = w_{old} + \Delta \mathrm{w}(\mathrm{t}) + \alpha \Delta \mathrm{w}(\mathrm{t} - 1)}$

Recurrent Neural Network



Elman Simple Recurrent Neural Network

To update the weights associated with input units coming from the output of hidden units:

$$\Delta w_{ij}(t) = \eta \delta_j net_j(t-1)$$

Linear Model: ARIMA

- ARIMA model is based on linear equation that consist of 3 parameters p, d, q
  - p = the number of autoregressive term,
  - d = the number of nonseasonal differences
  - q = the number of lagged forecast errors in the prediction equation
- "A common obstacle for many people in using Autoregressive Integrated Moving Average (ARIMA) models for forecasting is that the order selection process is usually considered subjective and difficult to apply".
- Hyndman and Khandakar suggest function auto.arima() that is already implemented in R. This function can determine the best value of p, d and q.

Outline

- Introduction
- **Project Objectives**
- Implementation
- Experimental Design
- Result
- Conclusion
- Future Work
- References

Objectives

- Compare the performance of time series forecasting using:
  - Feed Forward Neural Network (FFNN)
  - Recurrent Neural Network (RNN)
  - FFNN combined with Linear Method
  - RNN combined with Linear Method
- Hypothesis: RNN Hybrid is better than three other methods

Outline

- Introduction
- Project Objectives
- **Implementation**
- Experimental Design
- Result
- Conclusion
- Future Work
- References

+ Implementation

- Data set normalization.
  - min-max data normalization technique by Priddy and Keller in the range of [0.2, 0.8] based on Fishwick and Tang experiment
- Implementing FFNN for time series forecasting in Java.
- Implementing RNN for time series forecasting in Java.
  - RNN = Elman Simple Recurrent Neural Network
- Implementing FFNN Hybrid and RNN Hybrid in Java
  - Use Rcaller to call auto.arima() available in R.

FFNN Hybrid and RNN Hybrid

- Get neural network output value
- Calculate error = target – neural network output
- Use error as input for auto.arima()
- Get output of auto.arima()
- Compute forecast value of hybrid method:
  - Hybrid output = neural network output + auto.arima() output

Outline

- Introduction
- Project Objectives
- Implementation
- **Experimental Design**
- Result
- Conclusion
- Future Work
- References

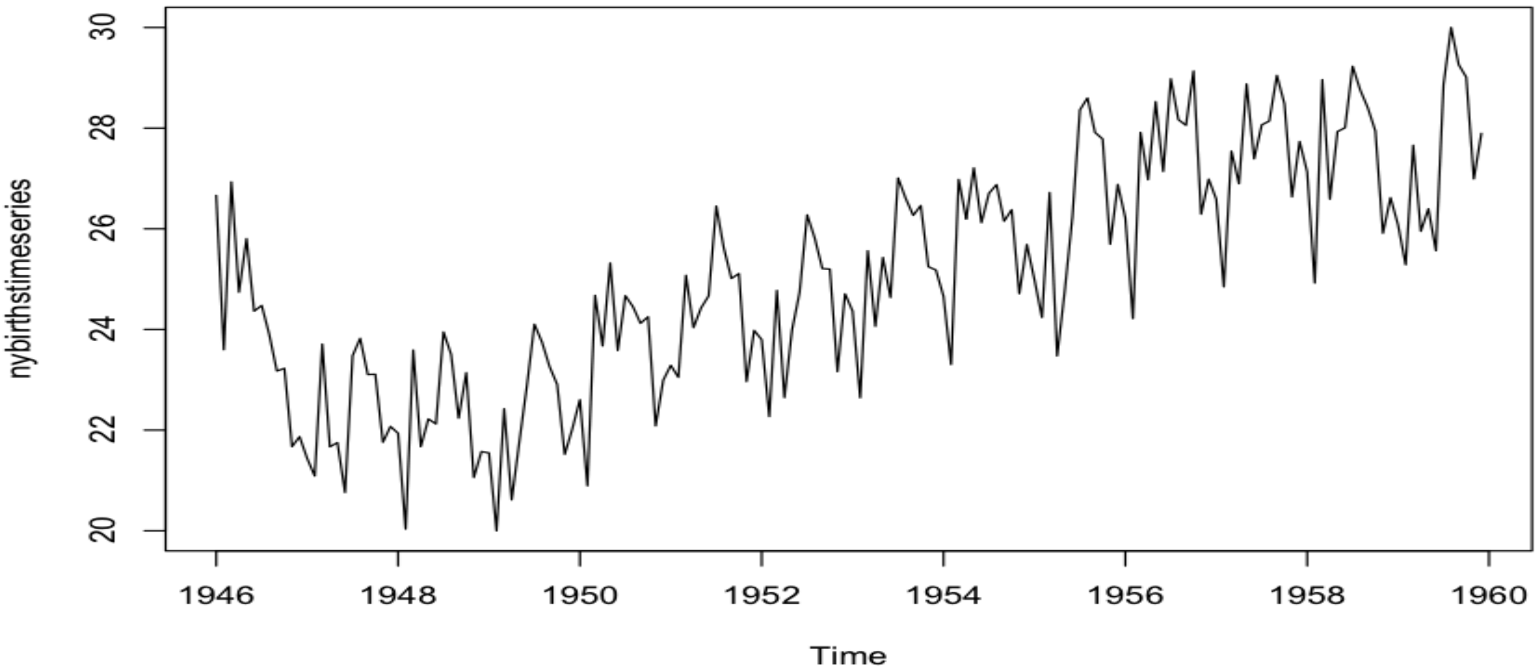
Experimental Design

- 3 data sets with different pattern:
  - New York Birth (NYB)
  - Milk Production
  - Skirt Diameter
- Run 30 trials on 7 different network topologies representing single step and multi step forecast
  - NYB and Milk Production: 1-6-1, 6-6-1, 6-6-4, 6-6-6, 12-6-1, 12-12-4, 12-18-12
  - Skirt Diameter: 1-5-1, 3-5-1, 3-5-3, 6-12-1, 6-12-4, 4-6-1, 6-4-3
- Initialize weights and bias between -0.5 and 0.5.
- Training Length = 500
- Learning Rate = 0.25

Experimental Design

- Apply t-test on these forecast errors for each network topology to find the best approach.
  - FFNN vs FFNN Hybrid
  - RNN vs RNN Hybrid
  - FFNN vs RNN
  - FFNN Hybrid vs RNN Hybrid

New York Birth Data Set

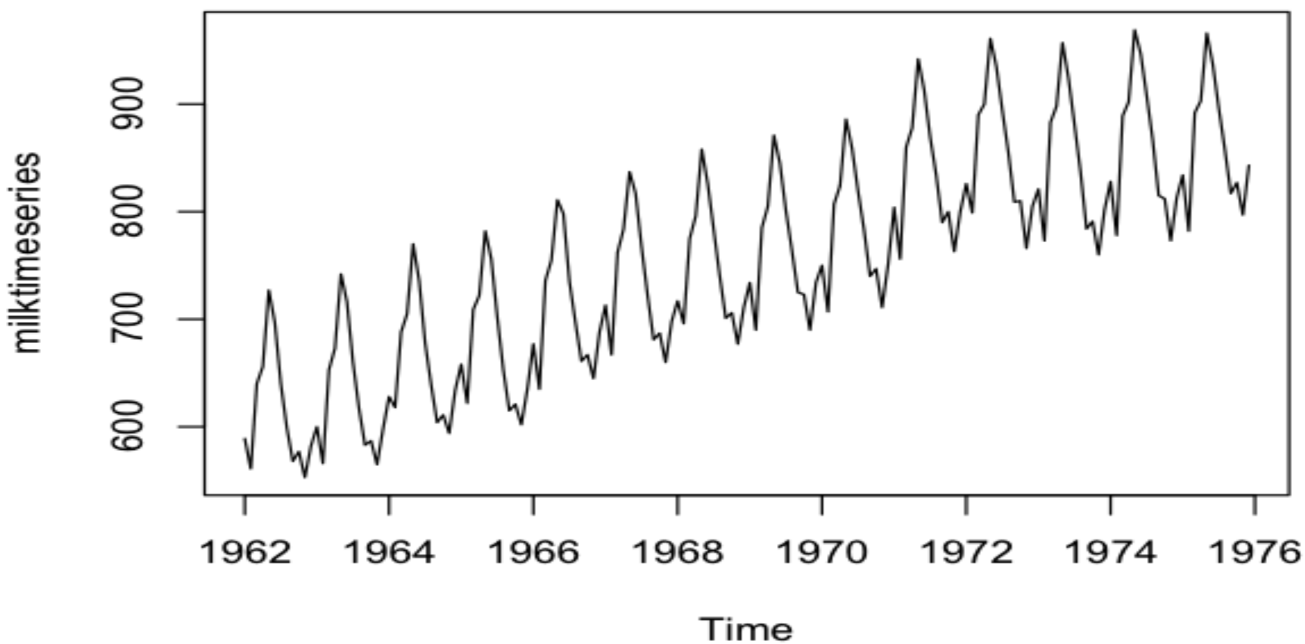


Data Training : 1946 - 1953
Data Testing : 1954 - 1956
Data Validation : 1957 – 1959

Sample RMSE, MAE, MAPE of NYB data set (1-6-1)

	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	V11	V12
1	1.990457	1.106632	1.327856	2.2770454	1.653406	0.9478784	1.059764	1.9116083	5.859068	3.438836	3.948910	7.179521
2	1.945280	1.091046	3.241550	1.2670305	1.614409	0.9302588	2.983457	1.0933388	5.723568	3.370950	10.654238	4.015975
3	1.961104	1.100805	1.913085	0.9521434	1.627011	0.9403619	1.600438	0.7946335	5.767361	3.409970	5.674902	2.841643
4	1.963422	1.045067	1.272518	1.5262884	1.626342	0.8856486	1.095491	1.1832871	5.765186	3.208358	4.027338	4.454446
5	1.992513	1.112678	1.316000	1.3028133	1.654716	0.9545978	1.148487	1.0664273	5.863785	3.465134	4.166038	3.959654
6	1.954417	1.043991	1.811610	0.9162291	1.618804	0.8861495	1.519067	0.7291134	5.738854	3.210070	5.392866	2.596633
7	1.966217	1.098665	1.386636	1.2831649	1.633025	0.9410680	1.192331	1.1254225	5.788116	3.412488	4.293900	4.106368
8	1.950216	1.097176	2.737195	1.2503583	1.618120	0.9370827	2.428677	1.0842345	5.736513	3.397124	8.634986	3.979339
9	1.961553	1.102036	1.383791	1.2755489	1.626968	0.9412953	1.187233	1.1123958	5.767170	3.413806	4.277079	4.069603
10	1.979095	1.055755	1.334812	1.2716590	1.641419	0.9012955	1.158456	1.1006509	5.817651	3.267797	4.192011	4.042900
11	1.994493	1.114625	1.349149	1.2746500	1.655288	0.9532202	1.164158	1.0902074	5.866080	3.460176	4.206635	4.015887
12	1.971971	1.107004	1.275772	1.5772831	1.636056	0.9477418	1.115689	1.2336705	5.798672	3.438882	4.079759	4.647532
13	1.973337	1.105714	1.287235	1.2737719	1.639602	0.9509839	1.126228	1.0882645	5.810923	3.450506	4.105856	4.008999
14	1.948383	1.098114	1.643154	0.9178029	1.616024	0.9382421	1.389796	0.7357347	5.729192	3.401880	4.950255	2.619650
15	1.977541	1.107397	1.344183	1.2770032	1.640962	0.9475981	1.161027	1.0834287	5.815829	3.438154	4.196892	3.997665
16	1.978039	1.099434	1.723771	0.9300695	1.645484	0.9463039	1.450634	0.7531295	5.831275	3.432061	5.156654	2.693938
17	1.968047	1.107043	1.490255	0.9716563	1.631609	0.9455404	1.273772	0.8005579	5.783348	3.430511	4.560173	2.884048
18	1.995363	1.116245	1.274906	1.6131854	1.655568	0.9544524	1.091249	1.2674570	5.867178	3.465044	4.019471	4.777015
19	1.980241	1.109707	1.301807	1.2730616	1.643287	0.9500594	1.138466	1.0907476	5.823974	3.447704	4.138845	4.016033
20	1.995697	1.113744	1.318799	1.2741804	1.656597	0.9529487	1.150943	1.0983033	5.870598	3.459048	4.173749	4.038730
21	1.967141	1.101783	2.631468	1.1941944	1.635125	0.9480452	2.319607	1.0495933	5.795348	3.439033	8.241453	3.839712
22	1.968405	1.101397	1.437609	1.2999928	1.635988	0.9464253	1.230211	1.1384232	5.798425	3.432658	4.415644	4.139531
23	1.953293	1.046005	2.196178	1.1214917	1.618014	0.8893066	1.866477	0.9882851	5.736158	3.221850	6.612255	3.594823
24	1.981218	1.107182	1.310848	1.2909312	1.646848	0.9540204	1.144910	1.0705958	5.835968	3.462257	4.155613	3.966669
25	1.958046	1.092503	1.765271	0.9194270	1.627489	0.9361097	1.482275	0.7437738	5.768900	3.392989	5.265382	2.648851
26	1.961410	1.098564	2.489725	1.1789432	1.630743	0.9455263	2.174644	1.0377198	5.780119	3.429219	7.720574	3.792176
27	1.951710	1.096424	1.697899	0.9543551	1.619984	0.9377591	1.429218	0.8048869	5.742887	3.399862	5.083784	2.879657
28	1.966780	1.102217	1.652816	0.9179738	1.634378	0.9478115	1.396027	0.7366948	5.792767	3.438260	4.971620	2.623237
29	1.969555	1.098372	1.872209	0.9327629	1.636851	0.9422989	1.578463	0.7779544	5.801399	3.416902	5.599108	2.776929
30	1.983328	1.107884	1.301708	1.2783487	1.648906	0.9550964	1.140806	1.0835402	5.843124	3.466263	4.147185	3.999110

Milk Production Data Set

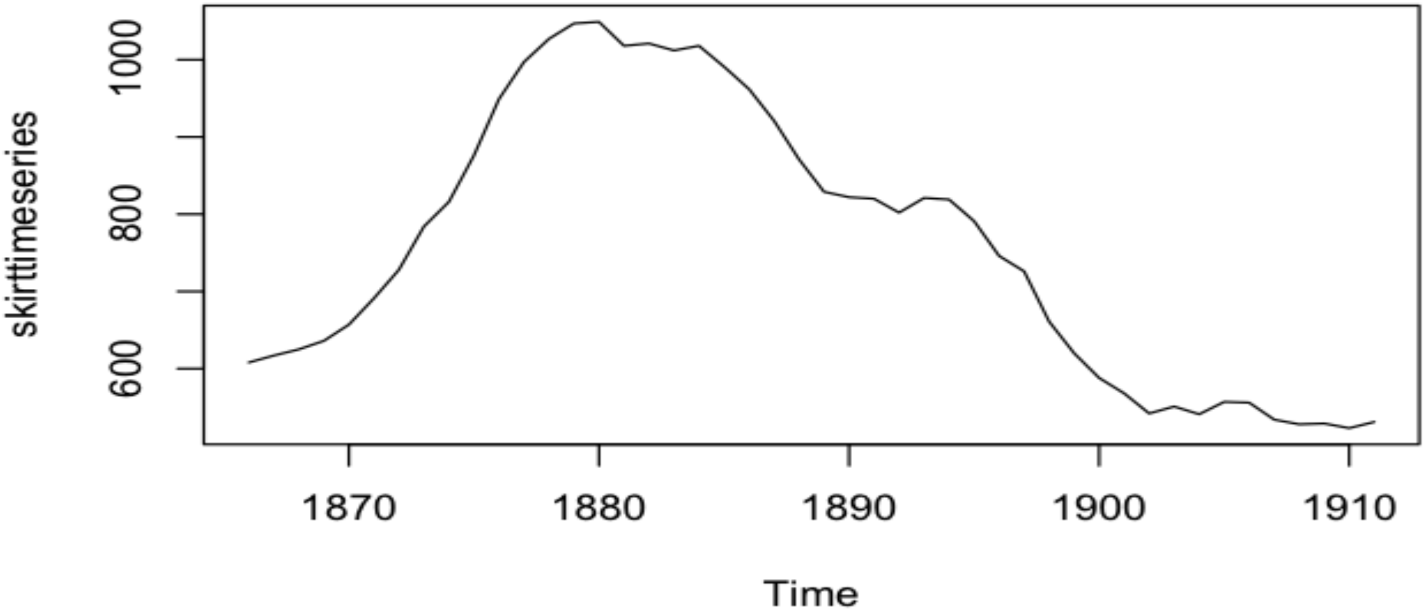


Data Training : 1962 - 1969
Data Testing : 1970 - 1972
Data Validation : 1973 - 1975

RMSE, MAE, MAPE of Milk Production data set (12-6-4)

	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	V11	V12
1	27.89283	12.08501	55.36375	126.26754	24.18778	10.139075	43.90601	105.99980	2.732682	1.1760624	5.232360	12.630238
2	26.61411	11.71230	54.26664	107.67610	22.83860	9.517730	44.90097	92.62618	2.579920	1.0888195	5.407430	11.117444
3	25.68620	10.96681	61.62655	133.45328	22.19072	9.104129	50.46159	97.67442	2.509762	1.0455923	5.958735	11.581304
4	29.65222	12.48633	53.23868	108.65431	26.35940	10.164803	44.21982	79.53533	2.996806	1.1791301	5.231254	9.448471
5	29.25851	11.64385	48.40450	93.08458	25.89433	9.577888	39.29833	77.94374	2.942071	1.1078791	4.710052	9.335186
6	24.15400	10.74570	67.73036	139.40328	21.25126	8.708082	58.05054	118.07444	2.423577	0.9991568	6.899044	13.990009
7	29.37958	12.16990	37.45850	89.26154	25.78704	10.281141	30.18771	75.28456	2.921886	1.1910052	3.596648	8.922887
8	31.23725	12.91406	39.53263	54.29339	27.28656	10.542210	31.48555	41.42228	3.086916	1.2166382	3.694987	4.974615
9	24.63692	11.20942	46.85087	105.49151	21.28794	9.019165	37.67650	94.75154	2.406717	1.0331952	4.534436	11.333092
10	27.68525	11.47926	54.60944	132.94074	23.71197	9.462973	44.23644	109.02831	2.670038	1.0846079	5.302282	12.969314
11	33.01254	12.20231	48.31151	100.97212	29.46825	9.833145	38.80175	60.37004	3.352012	1.1426566	4.488506	7.078428
12	31.29745	12.67541	67.37154	119.57910	27.48869	10.452503	55.65251	100.00681	3.109636	1.2085062	6.616975	11.928797
13	26.06455	11.81382	79.47073	168.49544	22.45219	9.630381	69.09746	156.03372	2.539083	1.1091068	8.296282	18.494178
14	23.03742	11.24448	43.27764	102.35971	19.38368	9.257542	35.52471	78.78445	2.191866	1.0572689	4.185262	9.392873
15	25.60510	11.23351	48.81640	128.92679	22.00432	8.902468	39.13718	108.82153	2.491274	1.0128658	4.719017	12.926704
16	25.70791	11.03346	41.14874	66.11414	22.46021	9.022238	31.82646	46.73706	2.545998	1.0401800	3.752216	5.556606
17	26.67049	12.26013	43.98354	82.29887	23.23001	10.068183	35.48985	63.48530	2.627261	1.1672171	4.240728	7.652545
18	25.92023	12.18890	58.90716	120.00020	22.50296	10.256656	47.33143	90.38330	2.542261	1.1796470	5.611232	10.740011
19	24.38703	11.21760	63.26669	114.71822	20.94693	9.328813	52.56522	97.27086	2.372079	1.0709706	6.301195	11.671948
20	23.94094	10.92966	63.00821	137.50748	20.59325	9.054065	52.13617	125.25057	2.326028	1.0384748	6.288229	14.912468
21	31.10605	11.89012	60.46186	142.09691	27.64042	9.890160	49.15575	124.59626	3.129185	1.1429492	5.894888	14.763445
22	28.47916	11.42493	45.65084	114.71328	24.75164	9.470236	35.92848	90.17743	2.796171	1.0867154	4.313463	10.687875
23	28.49784	12.14984	38.52083	43.52333	24.91002	10.002151	30.68134	32.99884	2.818666	1.1610201	3.593626	3.947757
24	29.74549	12.14476	74.05794	182.76781	26.01925	10.342090	62.43073	153.34855	2.939263	1.2045743	7.469129	18.107361
25	27.95243	12.57573	63.94076	139.73873	23.99328	9.979826	53.75890	121.19670	2.710675	1.1408553	6.478336	14.428355
26	25.73984	11.71182	48.14508	122.35353	22.38623	9.678681	39.56534	101.33087	2.537298	1.1088842	4.714873	12.045386
27	27.90041	12.70252	58.92789	130.95396	24.47745	10.756938	48.32413	109.65390	2.774892	1.2472116	5.825789	13.021675
28	25.17693	11.15841	50.83326	137.92328	21.73853	8.657775	40.82258	118.09203	2.462803	0.9808460	4.914479	14.023229
29	24.75990	10.91271	66.99636	132.95709	21.05061	8.977727	53.66258	107.84188	2.368638	1.0220161	6.474326	12.933070
30	27.68040	11.74173	42.61386	84.85099	24.20889	9.861261	34.29079	58.04163	2.742551	1.1402316	4.003047	6.876739

Skirt Diameter Data Set



Data Training : 1866 - 1889
Data Testing : 1890 - 1899
Data Validation : 1900 - 1910

+ RMSE, MAE, MAPE of Skirt Diameter data set (16-12-4)

	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	V11	V12
1	216.0178	379.5410	430.6119	836.5674	192.6889	330.3934	426.7492	832.0306	36.54484	62.66775	80.84849	157.6109
2	223.4728	411.1831	410.0243	768.9650	197.1017	361.1375	403.7881	749.1060	37.38330	68.49435	76.50608	141.9554
3	229.1500	424.4467	382.3749	682.6096	202.8298	372.6319	376.0134	663.3482	38.46806	70.67326	71.24771	125.7130
4	222.5204	390.7371	386.0204	741.7868	194.6689	334.9098	383.5407	736.9206	36.92395	63.52888	72.65333	139.5912
5	217.1292	394.1947	398.5568	713.4385	187.7466	335.9914	393.3029	705.2743	35.61250	63.73387	74.51396	133.6149
6	227.7648	423.4556	371.5796	668.8292	200.5371	370.3667	361.9797	642.1786	38.03497	70.24609	68.59797	121.7175
7	248.6703	420.6148	385.7643	699.8376	219.4026	368.0504	381.2731	683.8594	41.61479	69.80958	72.23087	129.5808
8	223.5088	411.3710	396.9028	737.6802	194.9723	354.2811	391.0019	720.4821	36.98152	67.20042	74.08280	136.5246
9	228.4404	424.1542	367.7118	564.6827	204.4136	377.1192	364.3182	548.9382	38.76629	71.51975	69.01407	104.0164
10	223.0383	406.3219	383.4990	735.8132	194.3540	350.6622	378.7154	722.4516	36.86506	66.51395	71.75208	136.8895
11	229.7245	408.7221	369.8127	681.9202	204.3917	359.0177	361.4109	665.9981	38.76187	68.08978	68.48327	126.1970
12	225.6756	418.7220	383.3512	668.7085	200.5230	367.4062	381.0828	659.6425	38.03046	69.68380	72.18784	124.9750
13	231.2380	423.0113	376.3845	673.5991	208.0747	377.9374	373.6188	665.4631	39.46112	71.67459	70.77269	126.0663
14	219.3440	396.9353	384.4580	672.8154	189.2555	339.4575	374.2883	647.1157	35.90108	64.39411	70.93015	122.6494
15	215.7325	400.8023	355.8480	565.1605	187.6781	344.0459	352.5975	556.2993	35.59988	65.26282	66.79758	105.4015
16	229.7861	423.5459	395.4589	703.9609	201.6421	367.3354	394.4322	700.6757	38.24573	69.67504	74.70742	132.7183
17	231.0427	414.7938	371.9043	708.0665	204.4750	364.4117	367.3324	699.0830	38.78026	69.11455	69.59788	132.4529
18	219.5314	400.4231	421.7890	822.0333	194.6111	350.7597	418.4153	810.6784	36.91094	66.52855	79.26581	153.5949
19	224.3722	415.0770	377.8096	616.2552	199.3601	366.1539	375.2576	607.1198	37.81087	69.44480	71.08358	115.0208
20	242.4881	427.6351	392.6312	713.4077	214.5750	376.0287	391.0960	707.4352	40.69755	71.31961	74.08161	134.0213
21	224.5538	410.6811	390.5160	653.6936	201.0663	363.1775	389.4174	648.4304	38.13078	68.87770	73.75716	122.8292
22	228.0588	399.9726	353.1946	601.0539	200.1890	345.3242	342.3181	559.5243	37.97062	65.50270	64.87393	106.0808
23	225.8779	416.6568	372.7037	693.1519	201.3040	366.6476	356.6962	652.4413	38.17757	69.53785	67.60986	123.6871
24	229.0226	422.6754	372.8338	665.6020	204.6303	373.1785	367.8845	644.7155	38.80830	70.77717	69.69597	122.1780
25	222.0239	409.5709	370.0352	687.3374	191.9663	351.0816	364.5268	675.4989	36.41327	66.59686	69.05956	127.9775
26	223.5346	410.7387	378.9159	754.2621	199.0177	361.2398	374.4338	743.4046	37.74548	68.51438	70.93621	140.8427
27	228.3548	420.7046	403.9927	778.4654	203.7227	370.3768	394.8502	755.4279	38.63589	70.24677	74.82214	143.1601
28	219.6315	400.5380	405.9814	764.1807	196.2742	351.7055	400.4954	746.4793	37.22416	66.70688	75.87923	141.4521
29	222.6175	390.0661	416.5860	834.3839	194.9163	335.4997	411.4048	827.9210	36.97185	63.64094	77.94382	156.8378
30	232.8834	425.3087	374.7922	670.6816	209.2197	380.6382	366.0394	651.6226	39.67777	72.18589	69.36371	123.4889

Outline

- Introduction
- Project Objectives
- Implementation
- Experimental Design
- **Result**
- Conclusion
- Future Work
- References

Result on NYB Data Set

	Network Topology (NYB data set)						
	1-6-1	6-6-1	6-6-4	6-6-6	12-6-1	12-12-4	12-18-12
FFNN vs FFNN Hybrid	FFFN Hybrid	FFFN Hybrid	FFFN Hybrid	FFFN Hybrid	FFFN Hybrid	FFFN Hybrid	FFFN Hybrid
RNN vs RNN Hybrid	RNN Hybrid	RNN Hybrid	RNN Hybrid	RNN Hybrid	RNN	RNN	RNN Hybrid
FFNN vs RNN	RNN	RNN	RNN	FFNN	FFNN	FFNN	FFNN
FFNN Hybrid vs RNN Hybrid	FFNN Hybrid	FFNN Hybrid	FFNN Hybrid	FFNN Hybrid	FFNN Hybrid	FFNN Hybrid	FFNN Hybrid

- The best approach : FFNN Hybrid

Result on Milk Production Data Set

	Network Topology (Milk Production data set)						
	1-6-1	6-6-1	6-6-4	6-6-6	12-6-1	12-12-4	12-18-12
FFNN vs FFNN Hybrid	FFFN Hybrid	FFFN Hybrid	FFFN Hybrid	FFFN Hybrid	FFFN Hybrid	FFFN Hybrid	FFFN Hybrid
RNN vs RNN Hybrid	RNN Hybrid	RNN Hybrid	RNN Hybrid	RNN Hybrid	RNN	RNN	RNN
FFNN vs RNN	FFNN	FFNN	FFNN	FFNN	FFNN	FFNN	FFNN
FFNN Hybrid vs RNN Hybrid	FFNN Hybrid	FFNN Hybrid	FFNN Hybrid	FFNN Hybrid	FFNN Hybrid	FFNN Hybrid	FFNN Hybrid

- The best approach : FFNN Hybrid

Result on Skirt Diameter Data Set

	Network Topology (Skirt Size Data Set)						
	1-5-1	3-5-1	3-5-3	16-12-1	6-12-4	4-6-1	6-4-3
FFNN vs FFNN Hybrid	FFFN Hybrid	FFFN Hybrid	FFFN Hybrid	FFFN Hybrid	FFFN Hybrid	FFFN Hybrid	FFFN Hybrid
RNN vs RNN Hybrid	RNN	RNN	RNN	RNN	RNN	RNN	RNN
FFNN vs RNN	FFNN	FFNN	FFNN	FFNN	FFNN	FFNN	FFNN
FFNN Hybrid vs RNN Hybrid	FFNN Hybrid	FFNN Hybrid	FFNN Hybrid	FFNN Hybrid	FFNN Hybrid	FFNN Hybrid	FFNN Hybrid

- The best approach : FFNN

Outline

- Introduction
- Project Objectives
- Implementation
- Experimental Design
- Result
- **Conclusion**
- Future Work
- References

Conclusion

- The performance of each method depends the characteristic of time series data and the network topology .
- FFNN Hybrid is better than FFNN, RNN and RNN Hybrid especially when the time series data contains trends and seasonal pattern.
- FFNN performs better than the other three methods when it is applied to irregular time series data

Conclusion

- If we need to shorten the computation time we can consider using regular FFNN or RNN.
- For time series data with irregular up and down movement:
  - RNN can outperform FFNN when they are applied for single step forecast with small number of input units.
  - RNN is also better than FNN when they are used for multistep forecast with small number of input and output unit.
- For time series with seasonal up and down movement:
  - FFNN is better than RNN

Outline

- Introduction
- Project Objectives
- Implementation
- Experimental Design
- Result
- Conclusion
- **Future Work**
- References

Future Works

- Expand the experiment by comparing the result from different epoch and learning parameter.
- Applying FFNN, FFNN Hybrid, RNN and RNN Hybrid to more various time series data is also necessary to substantiate the conclusion.
- Examine the performance of these 4 approaches when applied to step-wise forecast.
- Combine ARIMA with more sophisticated type of recurrent neural network and its learning algorithm

Outline

- Introduction
- Project Objectives
- Implementation
- Experimental Design
- Result
- Conclusion
- Future Work
- **References**

References

- Engelbrecht A. P., "Supervised Learning Neural Networks," in *Computational Intelligence An Introduction*, 2'nd ed: John Wiley and Sons, Ltd, 2007, pp. 27-54.
- Haykin S., "Dynamically Driven Recurrent Networks," in *Neural Networks and Learning Machines,* 3'rd ed New Jersey: Pearson Education, Inc., 2008, p 790-812
- Purwanto, Eswaran C. , and Longeswaran R., "An enhanced hybrid method for time series prediction using linear and neural network models," *Applied Intelligence*, vol. 37, pp. 511-519, 2012.
- Tang Z. and Fishwick P. A., "Feedforward Neural Nets as Models for Time Series Forecasting," *ORSA Journal on Computing*, vol. 5, pp. 374-385, 1993.
- Hyndman J. Rob and Khandakar Yeasmin, "Automatic Time Series Forecasting: The Forecast Package for R," *Journal of Statistical Software*, vol. 27, 2008.
- Priddy Kevin L. and Keller Paul E., "Min-Max Normalization," in *Artificial neural networks : an introduction*, Bellingham, Washington: SPIE- The International Society for Optical Engineering, 2005, p. 16.
- (05/01/2013). *Introduction to ARIMA: nonseasonal models*. Available: http://people.duke.edu/~rnau/411arim.htm