

UNIVERSIDAD NACIONAL SAN AGUSTÍN DE AREQUIPA

MAESTRIA EN CIENCIAS DE LA COMPUTACION



RECUPERACION DE LA INFORMACION

PROYECTO FINAL

TEMA: “Sectorización y detección de fugas de la red de agua potable de L-Town”

”

Docente:

Mg. ANA MARIA CUADROS VALDIVIA

Alumnos:

Gutiérrez Espinoza, Roger

Arequipa – 2023– I

Índice de Contenido

Sectorización y detección de fugas de la red de agua potable de L-Town.....	3
1. Introducción.....	3
2. Trabajos relacionados.....	3
3. Overview.....	4
4. Infraestructura	5
5. Análisis de tareas y/o requerimientos de diseño.....	6
6. Dataset.....	6
7. Metodología.....	7
a) Sectorización de la red L-Town	7
b) Detección de Fugas mediante una red neuronal profunda de 3 capas.	11
8. Conclusiones	13
9. Bibliografía.....	13

Índice de figuras

Figura 1. L-Town Areas.....	5
Figura 2. Localización de Recursos Hidráulicos	6
Figura 3. Cálculo de numero de Cluster con Método del Codo.....	10
Figura 4. L-Town sectorizado.....	10
Figura 5. Arquitectura Red Neuronal	11

Sectorización y detección de fugas de la red de agua potable de L-Town

1. Introducción

Las redes de distribución de agua potable son susceptibles a falla en su infraestructura, cual lleva a pérdida de agua. Estas pérdidas de agua se deben a fugas menores y ruptura de tuberías cual puede ocurrir en algún lugar dentro de la red de distribución, las fugas menores que están dentro de la red son difíciles de detectar, mientras que la ruptura de tuberías es más fácil de localizar por ser de gran tamaño y aparecen en la superficie.

La temprana detección y localización de fugas es importante porque reduciría el riesgo de una mayor degradación de la infraestructura, eventos de contaminación y quejas de los consumidores.

El documento está organizado en los siguiente: Parte 2 trata el desarrollo de los papers seleccionados para el presente trabajo , Parte 3 muestra la descripción del lugar de L-Town de donde se extrajo el dataset, Parte 6 presenta el análisis del dataset L-Town , la parte 7.1 muestra la metodología de sectorización de la red de la red de agua potable y sección 7.2 muestra un algoritmo no supervisado para la detección de fugas, en estas dos últimas secciones se analizara usando la red modelo de L-TOWN propuesto por Battle of the Leakage Detection and Isolation Methods (BattLeDIM) 2020 challenge. Finalmente, Parte 8 concluye este trabajo.

El trabajo se muestra en:

https://github.com/rogersisci/Recuperaciondeinformacion_unsa.git

El data set de la red de L-Town

es un pequeño pueblo con una población alrededor de 10 000 personas. La empresa de agua de L-Town es responsable de entregar agua potable a los consumidores a través de una red de tuberías con una longitud total de 42,6 km.

2. Trabajos relacionados

(Vrachimis, S. G et. al, 2020) BattLeDIM ha presentado la red L-Town que representa una pequeña ciudad hipotética con 782 nodos internos y dos

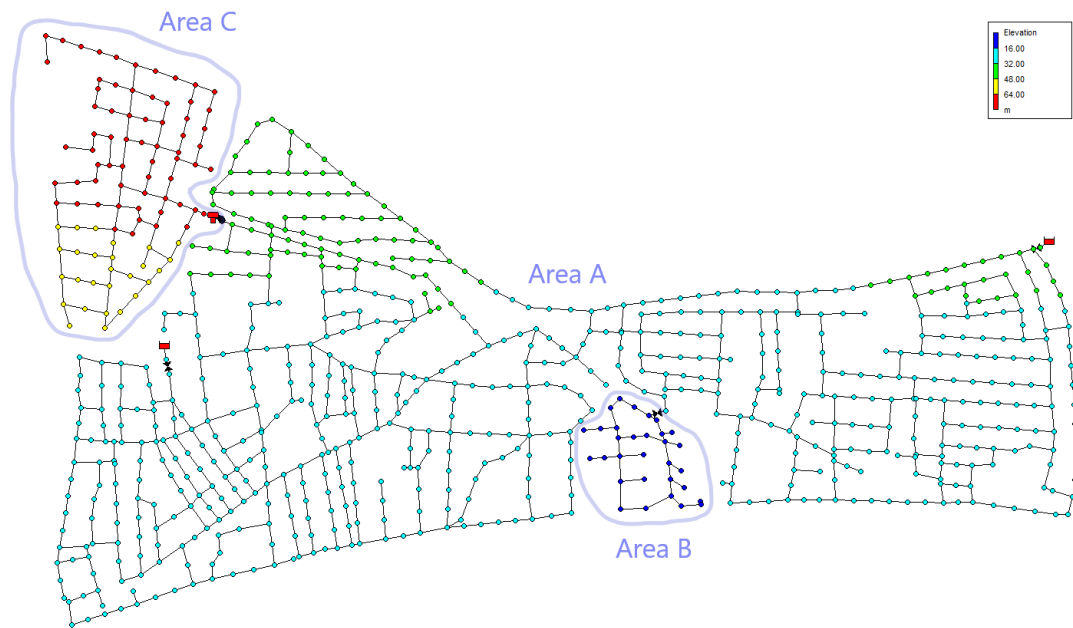
depósitos, un tanque. Se presenta un desafío de la rápida detección de fugas y el hecho de que el sistema tuviera múltiples fugas durante el año. La técnica puede dar una estimación de la magnitud de la fuga, y ante la presencia de una segunda fuga, la estimación es la suma de estas dos fugas, siendo necesaria la intervención humana para evaluar la presencia de una multifuga.

(Armando Di Nardo,2017) en Weighted spectral clustering for water distribution network partitioning muestra un algoritmo de clustering en DMA(district metric area) para redes de distribución de agua potable esto para la mejor detección de fugas de agua y localización de válvulas y sensores. Para lograr el objetivo usa el clustering spectral.

3. Overview

La red de distribución de agua de L-Town recibe agua de dos (2) embalses, y la empresa de agua tiene como objetivo proporcionar agua con una altura de presión de al menos 20 m a todos sus consumidores. Se instala una válvula de reducción de presión (PRV) en la parte baja de la ciudad ("Área B"), para ayudar a reducir las fugas de fondo. Las PRV también se instalan aguas abajo de los dos depósitos principales para ayudar a regular la presión. Se ha instalado una bomba y un depósito de agua en la parte alta del pueblo ("Zona C"), para dar suficiente presión a los consumidores de esa zona. El tanque tiene un diámetro de 16 metros con forma cilíndrica. La bomba ha sido programada para que el tanque se rellene durante la noche y se vacíe al "Área C" durante el día. Hay tres tipos de consumidores en L-Town: residencial, comercial e industrial. Durante los días laborables (lunes a viernes), el consumo de agua sigue un patrón similar, mientras que durante el fin de semana (sábado y domingo), se produce un mayor consumo en horas de la madrugada como consecuencia de la vida nocturna. Las áreas con usuarios industriales no siguen el mismo patrón de consumo. L-Town se encuentra en el hemisferio norte, por lo que se espera un uso de agua mayor alrededor de julio/agosto y menor en diciembre/enero. A efectos de este reto, no existen variaciones significativas en el consumo de agua durante las vacaciones u otros días especiales

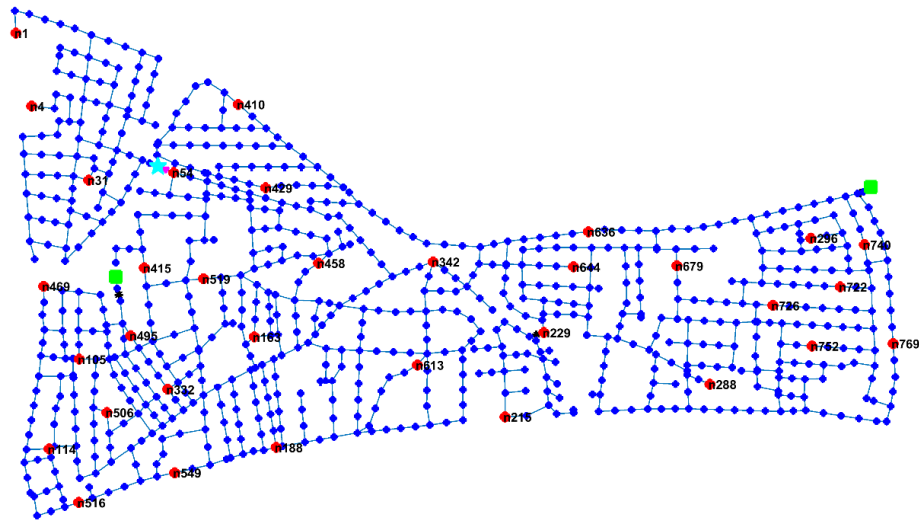
Figura 1. L-Town Areas



4. Infraestructura

Para mejorar su capacidad de monitorear pérdidas de agua, la empresa de agua de L-Town decidió instalar un (1) sensor de nivel de agua en el tanque, tres (3) sensores de flujo y 33 sensores de presión (ver Fig. 2), todos transmitiendo su medida -mensajes cada 5 minutos al Sistema de Supervisión, Control y Adquisición de Datos (SCADA) de la empresa de servicios públicos. Los sensores de presión dan un valor promedio de los últimos 5 minutos, lo que mitiga la incertidumbre por transitorios de presión en el sistema. Además, se han instalado 82 Lecturas Medidas Automatizadas (AMR) en el “Área C”, para entregar datos de consumo de agua directamente al sistema SCADA (ver Fig. 3). Cada AMR da el consumo agregado de muchos usuarios en el área AMR.

Figura 2. Localización de Recursos Hidráulicos



5. Análisis de tareas y/o requerimientos de diseño.

La empresa de servicios públicos de agua de L-Town está buscando una solución que les ayude a analizar el conjunto de datos SCADA y detectar eventos de fuga lo más rápido posible. Además, es crucial que la empresa de servicios públicos tenga una indicación de dónde ocurre aproximadamente la fuga, de modo que los trabajadores de campo puedan inspeccionar esas posibles fugas utilizando su equipo.

Para lograr la sectorización se realiza por medio de tratamiento de datos y clusterización Kmean.

Para la detección de fugas se realiza una Red Neuronal profunda de tres capas

6. Dataset

“L-Town.inp” EPANET Model (INP): Se proporciona un modelo de la red con parámetros para todos los elementos del sistema de red de agua. Las demandas base para cada nodo de la red se basan en el consumo promedio histórico medido. En general, la diferencia entre los valores reales y nominales para cada tipo de consumidor (residencial y comercial) es inferior al 10%.

También se proporcionan perfiles de demanda semanales para dos tipos de consumidores (residencial y comercial), sin embargo, no capturan la estacionalidad anual.

“2018 SCADA.xlsx” dataset historico (tambien en formato CSV): el dataset es una colección histórica del 2018-01-01 00:00 hasta 2018-12-31 23:55, en cada 5-minutos. SCADA dataset es comprimido del nivel del reserovior de agua, el flujo de los sensores , el AMR.

Detalle de Sensores:

- Existen tres (3) sensores de flujo en el sistema, dos en las entradas de DMA y uno en la bomba. Unidades de medida: metros cúbicos por hora (m^3/h).
- Existen 33 sensores de presión en el sistema que dan la presión promedio de los últimos 5 minutos en la ubicación del sensor. Los sensores de presión se colocaron utilizando una metodología que maximiza la sensibilidad colectiva de los sensores ante cualquier posible fuga. Unidades de medida: metros (m).
- Se ha instalado un (1) sensor de nivel de agua en el tanque. Unidades de medida: metros (m)
- 82 AMR instalados en “Área C” que dan el consumo agregado

7. Metodología

a) Sectorización de la red L-Town

Del dataset obtenemos las características de x,y,z,demanda y patrón

```
## importamos librerias
import pandas as pd
from sklearn.preprocessing import StandardScaler,MinMaxScaler
```

```
## lectura de datos
%run -i read_data.py
from read_data import read_inp_file
junctions, pipes, cordinates, patterns = read_inp_file("L-TOWN.inp")
dt=pd.merge(cordinates, junctions, left_on=';Node', right_on=';ID')
df=dt[['X-Coord','Y-Coord','Elev','Demand','Pattern']]
df
```

	X-Coord	Y-Coord	Elev	Demand	Pattern
0	138.22	1549.64	73.2105	0.000000	P-Residential
1	274.23	1504.56	73.8737	0.169920	P-Residential
2	271.28	1480.96	73.1782	0.190800	P-Residential
3	186.62	1326.65	68.2608	0.000000	P-Residential
4	158.93	1233.34	65.5695	0.232200	P-Residential
...
777	1487.17	801.53	27.9200	0.000000	P-Residential
778	1452.46	826.63	27.6400	0.143530	P-Residential
779	1487.84	761.07	26.4000	0.100840	P-Residential
780	1440.88	763.00	24.6000	0.034200	P-Residential
781	1404.56	807.26	25.0800	0.101630	P-Residential

782 rows × 5 columns

Realizamos el escalado de las características

```
## Escalado de datos
scaler = MinMaxScaler()
df_std = scaler.fit_transform(df)
df_std = pd.DataFrame(df_std, columns=df.columns)
## verificamos la estandarizacion
print(df_std.describe())
df_std
```

	X-Coord	Y-Coord	Elev	Demand
0	0.001185	0.955849	0.960849	0.000000
1	0.051551	0.926588	0.969782	0.170729
2	0.050459	0.911270	0.960414	0.191709
3	0.019108	0.811110	0.894183	0.000000
4	0.008854	0.750544	0.857935	0.233306
...
777	0.500720	0.470262	0.350844	0.000000
778	0.487867	0.486554	0.347073	0.144214
779	0.500968	0.444000	0.330372	0.101320
780	0.483578	0.445253	0.306128	0.034363

781	0.470129	0.473981	0.312593	0.102114
-----	----------	----------	----------	----------

782 rows × 4 columns

Realizamos la reducción de dimensionalidad PCA:

```
from sklearn.decomposition import PCA
pca = PCA(n_components=0.9)
pca.fit(df_std)
df_pca = pca.transform(df_std)
print(f"Original data: {df_std.shape}")
print(f"data Transformadas: {df_pca.shape}")
```

Original data: (782, 4)

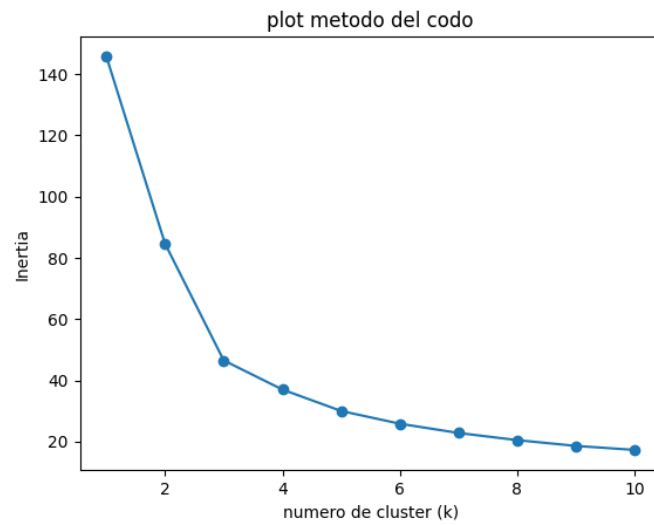
data Transformadas: (782, 3)

La original data (df_std) tiene 782 filas y 4 columnas, mientras la data transformada (df_pca) tiene 618 filas y 3 columnas. Esto significa que realizar la reducción de funciones de PCA redujo la cantidad de funciones de 4 a 3, al mismo tiempo que se conservó el 90 % de la varianza explicada en los datos.

Calculamos el número de Cluster para la red L-Town

```
from sklearn.cluster import KMeans
import matplotlib.pyplot as plt
inertia = []
for k in range(1, 11):
    kmeans = KMeans(n_clusters=k, random_state=42)
    kmeans.fit(df_pca)
    inertia.append(kmeans.inertia_)
plt.plot(range(1, 11), inertia, marker='o')
plt.xlabel('Numero de cluster (k)')
plt.ylabel('Inertia')
plt.title('metodo de codo plot')
plt.show()
```

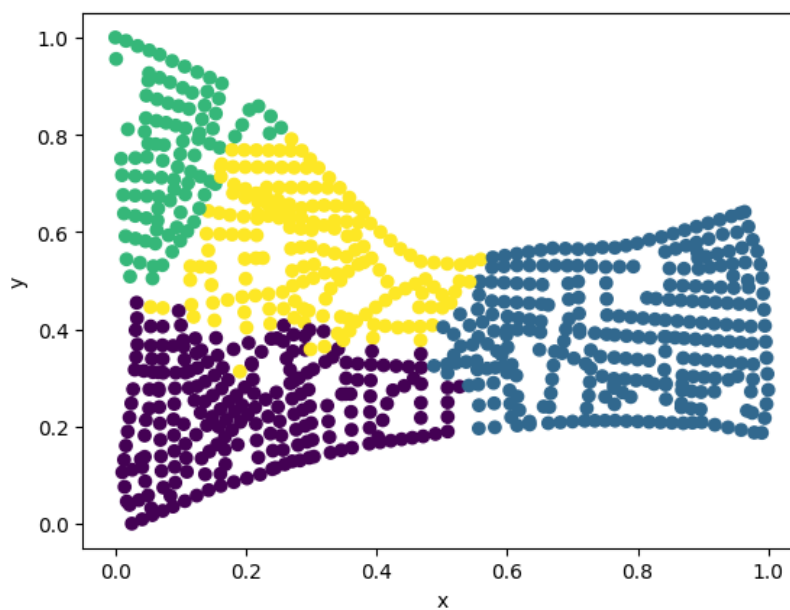
Figura 3. Cálculo de numero de Cluster con Método del Codo



Ploteamos la red sectorizada

```
km = KMeans(n_clusters=4)
km.fit(df_pca)
label=km.labels_
label
plt.scatter(df_std['X-Coord'], df_std['Y-Coord'], c=label)
plt.xlabel('X')
plt.ylabel('Y')
plt.show()
```

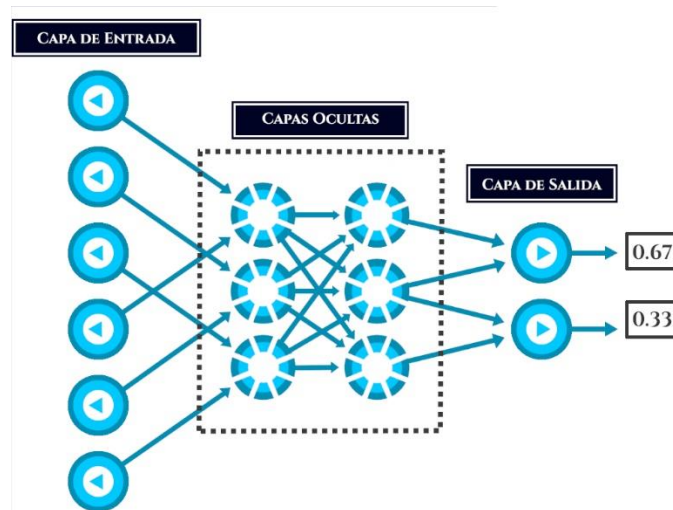
Figura 4. L-Town sectorizado



b) Detección de Fugas mediante una red neuronal profunda de 3 capas.

Cada modelo presenta una capa de entrada tenemos las presiones en los nodos donde están ubicados los sensores, tenemos 02 capas ocultas; y 02 capa de salida.

Figura 5. Arquitectura Red Neuronal



- Leemos todos los archivos de entrada del Dataset

```
pressures, demands, flows, levels, leakages = read_file("Database/2018_SCADA")
junctions, pipes, cordinales, patterns = read_inp_file("Database/L-TOWN.inp")
data = water_network(pressures, demands, flows, levels, junctions,
pipes, leakages, cordinales)
```

- Realizamos el preprocesamiento, escalamiento para el entrenamiento

```
pressures = data.presses.iloc[:,1:].to_numpy()
labels = np.sum(leakages.iloc[:, 1:].to_numpy(), axis=1)
labels = [0 if i==0 else 1 for i in labels]
x_train, x_test, y_train, y_test = data_preprocessing(presses, labels)
```

- Inicializamos el modelo

```
layer_sizes = [30,50,70,100,300]
```

```
input_shape = (np.shape(x_train)[1],)
```

```
models= models_init(2,layer_sizes,input_shape,'relu','sigmoid')
```

- entrenamos los modelos

```
models_f, histories = models_fit(x_train, y_train, models)
```

- evaluamos los modelos

```
models_f, scores = models_test(models_f,x_test,y_test, files_name = files_name)
```

- Resultado

	models_name	TN	FP	FN	TP	accuracy_score	loss_score	precision	recall	f1_score
0	(30, 30)	412	164	227	25477	0.985122	0.031619	0.715278	0.991169	0.830921
1	(30, 50)	414	162	262	25442	0.983866	0.031841	0.718750	0.989807	0.832777
2	(30, 70)	215	361	82	25622	0.983143	0.031798	0.373264	0.996810	0.543143
3	(30, 100)	487	89	464	25240	0.978957	0.044673	0.845486	0.981948	0.908622
4	(30, 300)	396	180	214	25490	0.985008	0.031328	0.687500	0.991674	0.812037
5	(50, 30)	325	251	192	25512	0.983143	0.034347	0.564236	0.992530	0.719468
6	(50, 50)	401	175	223	25481	0.984855	0.032409	0.696181	0.991324	0.817942
7	(50, 70)	240	336	78	25626	0.984247	0.030943	0.416667	0.996965	0.587709
8	(50, 100)	239	337	90	25614	0.983752	0.031353	0.414931	0.996499	0.585899
9	(50, 300)	250	326	85	25619	0.984361	0.030886	0.434028	0.996693	0.604720
10	(70, 30)	379	197	204	25500	0.984741	0.031847	0.657986	0.992063	0.791205
11	(70, 50)	324	252	176	25528	0.983714	0.036414	0.562500	0.993153	0.718217
12	(70, 70)	328	248	158	25546	0.984551	0.031044	0.569444	0.993853	0.724039
13	(70, 100)	381	195	179	25525	0.985769	0.031248	0.661458	0.993036	0.794021
14	(70, 300)	355	221	241	25463	0.982420	0.035011	0.616319	0.990624	0.759878
15	(100, 30)	371	205	155	25549	0.986301	0.031020	0.644097	0.993970	0.781669
16	(100, 50)	229	347	87	25617	0.983486	0.031165	0.397569	0.996615	0.568395
17	(100, 70)	128	448	41	25663	0.981393	0.036661	0.222222	0.998405	0.363531
18	(100, 100)	358	218	199	25505	0.984132	0.031960	0.621528	0.992258	0.764310
19	(100, 300)	215	361	73	25631	0.983486	0.032018	0.373264	0.997160	0.543195
20	(300, 30)	291	285	122	25582	0.984513	0.030954	0.505208	0.995254	0.670208
21	(300, 50)	307	269	109	25595	0.985616	0.030869	0.532986	0.995759	0.694329
22	(300, 70)	304	272	129	25575	0.984741	0.031436	0.527778	0.994981	0.689707
23	(300, 100)	307	269	138	25566	0.984513	0.031528	0.532986	0.994631	0.694054
24	(300, 300)	233	343	69	25635	0.984323	0.031768	0.404514	0.997316	0.575574

8. Conclusiones

- El análisis de clustering con las características topográficas x,y,z y patrones de demanda por nodos nos muestra una sectorización de 03 DMA.
- En el análisis de fugas se visualiza con un acuraccy más del 90% y próximo a los datos Test lo cual es adecuado

9. Bibliografía

- Di Nardo, A., Di Natale, M., Giudicianni, C. *et al.* Weighted spectral clustering for water distribution network partitioning. *Appl Netw Sci* **2**, 19 (2017). <https://doi.org/10.1007/s41109-017-0033-4>.
- Vrachimis, S. G., Eliades, D. G., Taormina, R., Ostfeld, A., Kapelan, Z., Liu, S., ... & Polycarpou, M. M.(2020). BattLeDIM: Battle of the Leakage Detection and Isolation Methods.
- Vrachimis, S. G., Kyriakou, M. S., Eliades, D. G. and Polycarpou, M. M. (2018). LeakDB: A benchmark dataset for leakage diagnosis in water distribution networks. In Proc. of WDSA / CCWI Joint Conference (Vol. 1).