Water Reuse

Water Reuse Vol 13 No 1, 51 doi: 10.2166/wrd.2022.069

Using machine learning architecture to optimize and model the treatment process for saline water level analysis

Sarvesh P. S. Rajput[a], Julian L. Webber[b], Ali Bostani[c], Abolfazl Mehbodniya[b], Mahendran Arumugam[d], Preethi Nanjundan[e] and Adimas Wendimagegen[f,*]

[a] Department of Civil Engineering, Maulana Azad National Institute of Technology, Bhopal, MP, India
[b] Department of Electronics and Communication Engineering, Kuwait College of Science and Technology (KCST), 7th Ring Road, Doha Area, Kuwait
[c] College of Engineering and Applied Sciences, American University of Kuwait, Salmiya, Kuwait
[d] Center for Transdisciplinary Research, Saveetha Dental College, Saveetha Institute of Medical and Technical Science, Chennai, India
[e] Department of Data Science, Christ University, Pune, Lavasa, Maharashtra, India
[f] College of Natural and Computational Science, Debre Berhan University, Debre Birha, Ethiopia
*Corresponding author. E-mail: adimaswendimagegen@dbu.edu.et

ABSTRACT

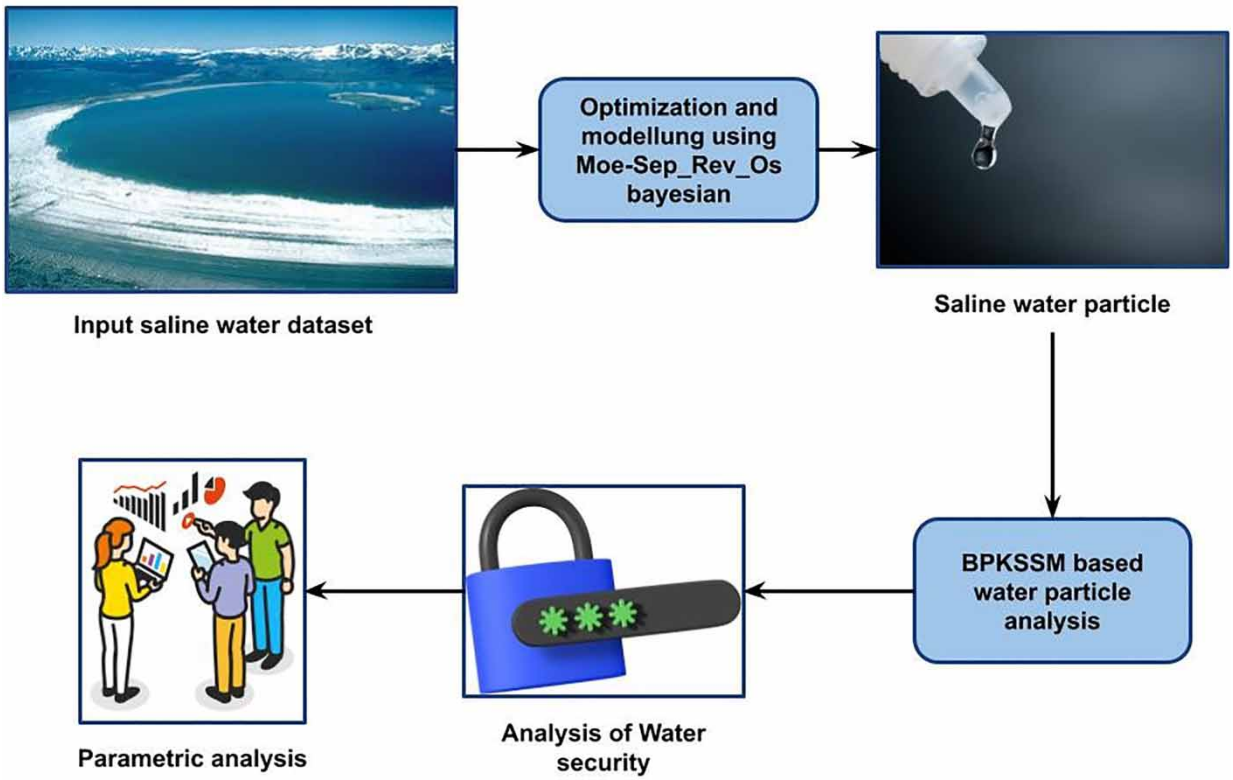
Water is a vital resource that makes it possible for human life forms to exist. The need for freshwater consumption has significantly increased in recent years. Seawater treatment facilities are less dependable and efficient. Deep learning systems have the potential to increase the efficiency as well as the accuracy of salt particle analysis in saltwater, which will benefit water treatment plant performance. This research proposed a novel method for optimization and modelling of the treatment process for saline water based on water level data analysis using machine learning (ML) techniques. Here, the optimization and modelling are carried out using molecular separation-based reverse osmosis Bayesian optimization. Then the modelled water saline particle analysis has been carried out using back propagation with Kernelized support swarm machine. Experimental analysis is carried out based on water salinity data in terms of accuracy, precision, recall, and specificity, computational cost, and Kappa coefficient. The proposed technique attained an accuracy of 92%, precision of 83%, recall of 78%, specificity of 81%, computational cost of 59%, and Kappa coefficient of 78%.

Key words: machine learning, saline water, water level data analysis, water saline particle, water treatment plants

HIGHLIGHTS

- A novel method in optimization and modelling of the treatment process for saline water based on water level data analysis using machine learning techniques is proposed.
- Here the optimization and modelling are carried out using molecular separation-based reverse osmosis Bayesian optimization.
- The modelled water saline particle analysis has been carried out using back propagation with a Kernelized support swarm machine.

GRAPHICAL ABSTRACT



1. INTRODUCTION

Despite the fact that 75% of the planet is covered by water bodies, a catastrophic water scarcity that the world faces in proportion to population growth, industrial development, and agricultural advancement has left 800 million people without access to drinking water. As 97% of the planet's water supplies are salty and the majority of freshwater is in glaciers, ice caps, and underground, there is also sadly very little drinking water available. By 2050, it is anticipated that roughly half of the world's freshwater would be utilized (Alshehri *et al.* 2021). Additionally, researchers have considered using a variety of water desalination methods to effectively combat this issue. Numerous efficient desalination techniques have been researched, including vapour compression, multistage flashing, reverse osmosis, and multi-effect distillation (Aghilesh *et al.* 2021). These techniques require high-power sources like fuel and energy, expensive installation and maintenance costs, and highly skilled operators and employees despite being efficient and capacitive methods on a big production scale. Desalination and water treatment have both made extensive use of the electric ion separation technique known as capacitive deionization (CDI) (Jiang *et al.* 2021). In order to remove salt from the electrodes, the CDI process involves repeated cycles of charging (ion adsorption) as well as discharging, which releases the salt into brine. During the charging phase of the CDI process, ions from the feed water are electrically adsorbed to an electrode (which is mostly made of carbon), with cations adhering to cathode as well as anions adhering to anode. As a result, the electrode's adsorption capacity and amount of ion adsorption of CDI are strongly connected. In order to discharge effectively and completely utilize this capacity, co-ions (ions with the same polarity as the electrode) must not desorb. A membrane CDI (MCDI) approach is developed based on the use of an ion exchange membrane (IEM) electrode surface to solve the co-ion desorption problem (Sahour *et al.* 2020). Ion adsorption effectiveness of MCDI is higher than that of CDI because it can successfully prevent the desorption of co-ions caused by the presence of IEMs.

Emerging Artificial Intelligence (AI) and Machine learning (ML), coupled with smart technology, are filling a niche in water applications that were previously underserved by traditional techniques and thinking. Some reports estimate that AI expenditures in the water industry will account for almost 10% of the investment of over $90 billion that is expected to

mature by 2030. In water applications, AI, ML, and smart technologies are expected to model and overcome complex and difficult issues through their generalization, resilience, and relative ease of design to achieve cost savings and optimize processes (Odabaşı *et al.* 2022). Water applications that have seen notable ML utilization include water and wastewater treatment, natural-systems monitoring, and precision/water-based agriculture. These industry studies have been observed to rely on numerous ML techniques, with the most commonly used including artificial neural networks (ANNs), recurrent neural networks (RNN), radio frequency (RF), support vector machine (SVM), and adaptive neuro fuzzy inference system (ANFIS) with occasional AI methods including fuzzy inference systems (FISs). There have also been some applications involving hybrid techniques that marry two ML systems, including ANN–RF and SVM–RF. Studies have recorded success in their applications of both AI and ML in water-based usages for optimizing modelling processes (Rall *et al.* 2020).

The contribution of this research is as follows:

1. To propose a novel method for the optimization and modelling of the treatment process for saline water based on water level data analysis using ML techniques.
2. Here the optimization and modelling are carried out using molecular separation-based reverse osmosis Bayesian optimization.
3. The modelled water saline particle analysis has been carried out using back propagation with a Kernelized support swarm machine.

The organization of this article is as follows: section 2 gives existing works based on optimization and modelling for saline water using ML techniques; section 3 explains proposed optimization and modelling for saline water using ML architectures; performance analysis and experimentation is shown in section 4; and we conclude in section 5 with future scope for research.

2. RELATED WORKS

Researchers from all around the world have suggested and presented a variety of approaches to forecast salinity and dangerous components in saline water. The following discussion includes a few of the suggested best-in-class methods. A hybrid strategy was put up in Work (Zhang *et al.* 2022) to forecast river water salinity in Babol-Rood River. ML methods input variables were predicted utilizing Pearson's correlation coefficient technique. Total Dissolved Solids (TDS) were also the primary predictor utilized to forecast the salinity of river water. Zhang *et al.* (2022) used an ANN with rapid propagation to predict the salinity forecast of groundwater depending on the pumping rate. A comparison between the implemented model and the standard statistical saturated-unsaturated (SUTRA) computational method was made. In the study (Viet *et al.* 2021), ANNs were used to evaluate groundwater salinity (ANN). Based on their findings, Bonny *et al.* (2022) and Lee *et al.* (2022) developed an image analysis technique to better understand how pollutant movement in homogenous aquifers is impacted by intrusive saltwater (SW). A linear relationship between concentration and dye optical density was established using nine distinct seawater concentrations. Wang *et al.* (2020) suggested an image analysis method that used a second-order exponential function of measured light intensity values to evaluate saltwater concentration in a homogeneous aquifer. The technique was further tested in two-dimensional heterogeneous aquifers, producing contour plots of intruding wedges' saltwater concentration with two (Li *et al.* 2020) and four (Doña *et al.* 2016) isolines, respectively. Feizizadeh *et al.* (2021) used a similar strategy, fitting eight salt concentration values to observed grayscale pixel intensity values using the least squares method. By mimicking an unrestricted coastal aquifer with quartz sand and using binary-style automated image processing, Batchuluun *et al.* (2021) reported the sole study to categorize each pixel into either freshwater or saltwater depending on its brightness levels. Lighting was positioned in front of the sandbox in every other study (light reflection technique). Son *et al.* (2021) made an effort to compare the two experimental strategies in-depth. The use of neural networks in membrane formation (Dargam *et al.* 2020) and the prediction of membrane function (Milosavljević 2020) are topics covered in a number of papers in the field of membrane science. By establishing superior membrane synthesis techniques based on the delicate balance between ion retention properties as well as permeability, we have extended the use of ANN surrogate methods in membrane research to characterize and optimize LbL-based membrane methods (Sagastibeltza *et al.* 2022). The performance of LbL nano-filtration membrane models, as well as separation method, may thus be designed simultaneously using an optimization technique thanks to the coupling of ANNs into a hybrid mechanistic/data-driven method. As a result, the intended separation goal can be best served by membrane synthesis techniques and membrane processes (Elsheikh *et al.* 2022).

3. SYSTEM MODEL

This section discusses a novel technique for the optimization and modelling of the treatment process for saline water based on water level data analysis using ML techniques. Here, the optimization and modelling are carried out using molecular separation-based reverse osmosis Bayesian optimization. Then the modelled water saline particle analysis has been carried out using back propagation with a Kernelized support swarm machine. The proposed architecture is shown in Figure 1.

We employed the Cook's distance to conduct the influential analysis in this study. The latter has been done to identify the influence of each data point on the final method and we also employed standard scaling. Data inside each feature are scaled by Standard Scaler so that distribution is centred around 0 and has a standard deviation of 1. Three continuous factors and one categorical component were taken into account throughout the experiment design (electrodes configuration). 45 trials were conducted using the Box-Behnken design with three iterations of a central point. For each experiment, the applied voltage required to sustain current intensity was monitored over the course of time (six sample points). Further, marginal distribution (the probability distribution of the variables contained in the subset) and Bayesian network inference (evaluating the joint probability of a particular assignment of values for each variable (or a subset) in the network) with naive Bayes (most effective classification algorithms which help in building the fast ML models that can make quick predictions) were estimated.

3.1. Molecular separation-based reverse osmosis Bayesian optimization

With an accuracy of $\pm$ 0.1 °C, a digital thermometer was used to track all temperatures. Additionally, a fan-type anemometer with a range of 0 to 45 m/s and an accuracy of 1 m/s was utilized to measure Vw. Additionally, IR was measured using a TES-1333 solar meter with a range of 0 to 5,000 W/m^2 and an accuracy of 5 W/m^2. Figure 2 includes a tubular cover, basin, feedwater assembly, and all necessary instrumentation for measurements of all specifications. As a tubular cover, a 1.5-thick, transparent polycarbonate cylinder measuring 50 cm in diameter and 100 cm in length was employed, allowing the basin to be exposed to sunlight from all directions. The basin also had side walls that measured 90, 40, 5, and 0.15 cm in length, breadth, and thickness, respectively, of black-painted steel. Through a feeding method that delivered feed water

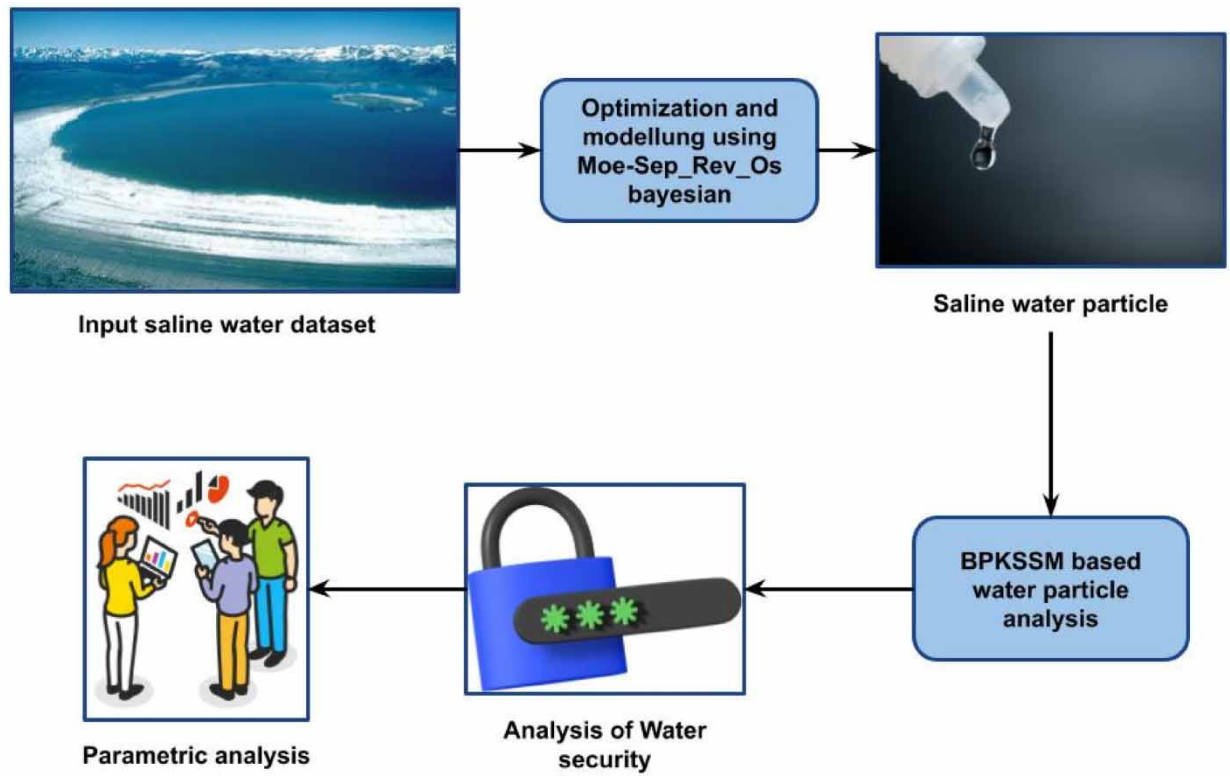


Figure 1 | Proposed architecture.

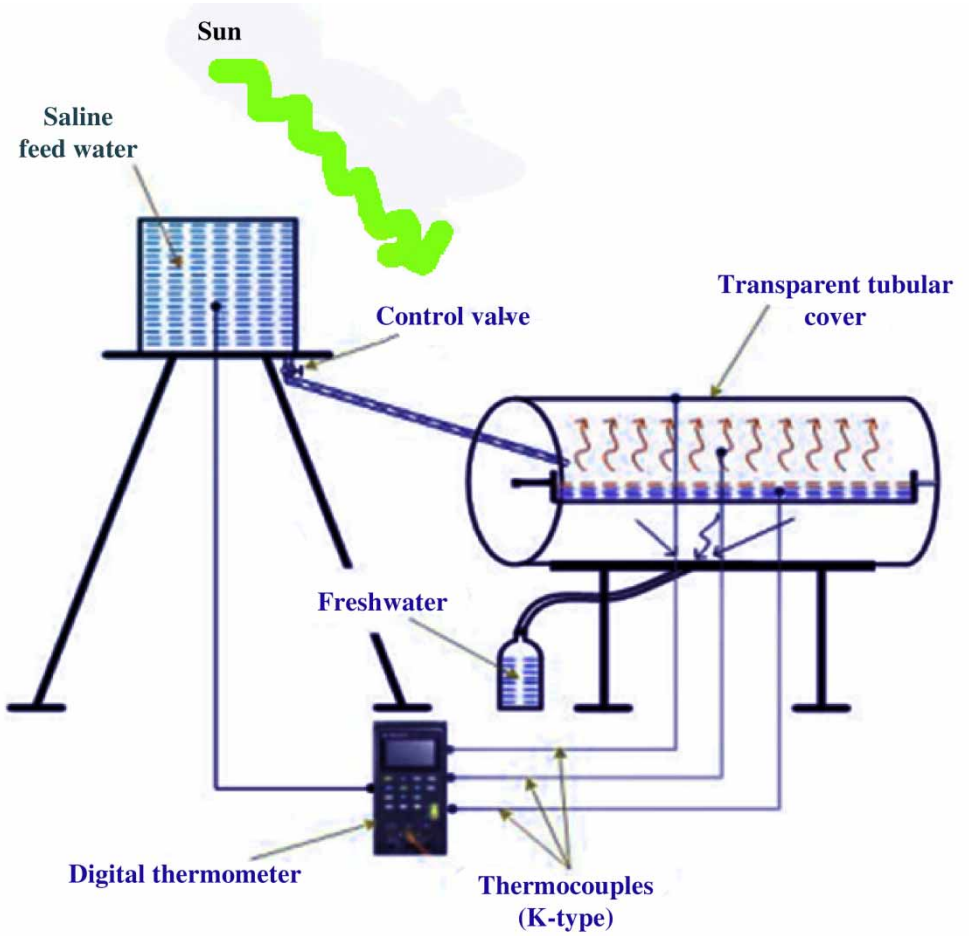


Figure 2 | Schematic diagram of molecular separation-based reverse osmosis.

at the same rate as hourly-evaporated amount, the height of the salinized water inside the basin was restricted to 0.5 cm. The main method used to determine this height was a graded level meter that was fastened to each side of the basin.

The apparatus was at room temperature (25 ± 2 °C), and the humidifier's temperature was set at 40 °C. He was fed through to eliminate the vapour from the system before the test. As a starting point for the NPP, steady permeance of dry gas was measured. Then, the vapour pressure was gradually raised by adjusting the mass flow controllers for dry gas (MF-1) and wet gas (MF-2) until capillary condensation or swelling caused by adsorption prevented gas permeation.

$$p = \frac{Q_w}{Q_w + Q_d} p_s \tag{1}$$

Equation (2) was utilized to find gas permeance, P_i:

$$P_i = \frac{Q_i}{\Delta p A} \tag{2}$$

where A is the membrane area and Q_i is the permeating flow rate. According to the Kelvin Equation (3), vapour sorption and condensation take place in small free-volume pores (diameter d_p) at vapour pressures p that are lower than P_s.

$$RT \ln\left(\frac{P}{P_s}\right) = 2v \frac{\sigma \cos\theta}{r_k} \tag{3}$$

where molar volume, surface tension, and contact angle are represented by υ, σ, and θ. The Kelvin radius, r_k, may be calculated using this equation. It is crucial to remember at this point that the uptake of a contaminant by two material adsorbents must be compared under identical experimental conditions as well as at the same equilibrium concentration (particularly pH). Separation factor, which is given by Equations (4) and (5) value of RL, determines whether the shape of isotherms is irreversible, favourable, linear, or unfavourable.

$$q_e = \frac{x}{m} = \frac{K_L C_e}{1 + a_L C_e} \tag{4}$$

$$R_L = \frac{1}{1 + a_L C_0} \tag{5}$$

The range of 1/nFs value between 0 and 1 reflects how nonlinearly the relationship between solution concentration and adsorption is. The adsorption process is chemical if the value of 1/nF is less than unity, linear if the value is equal to unity, and advantageously physical if the value is above unity. The more heterogeneous the surface, the closer the 1/nF value is to zero by Equation (6).

$$q_e = K_F C_e^{1/n_F} \tag{6}$$

There are conditional probability tables whose number of elements is exponential in n because a Bayesian network linked to G_i has a maximum in-degree (i.e., maximum number of parents) equal to n classes. For issues with numerous classes, this is not viable. Thus, the maximum in-degree can be decreased using a structural learning approach. In practice, we analyse the result of the following optimization to reduce certain classto-feature arcs in G_i: $G_i^* = \arg^{\mathrm{mnx}}_{G,CCD,} \log P(\mathcal{G} \mid \mathcal{D})$, where it is anticipated that set inclusions between graphs take place in the arcs space and are given by Equation (7)

$$\log P(\mathcal{G} \mid \mathcal{D}) = \sum_{i=1}^{n} \psi_a[C_i, \mathrm{Pa}(C_i)] + \sum_{j=1}^{m} \psi_a[F_j, \mathrm{Pa}(F_j)] \tag{7}$$

where $P_a(F_j)$ stands for F_j's parents according to G, and $P_a(C_i)$ means C_i's parents. Additionally, ψ_α BDEu score has a similar sample size α. For instance, Equation (8) score

$$\psi\alpha[F_j, P_a(\mathrm{Fj})]\mathrm{is} \sum_{i=1}^{|\mathrm{Pa}(F_i)|} \left[\log \frac{\Gamma(\alpha_j)}{\Gamma(\alpha_j + N_{ji})} + \sum_{k=1}^{|F_j|} \log \frac{\Gamma(\alpha_{ji} + N_{jik})}{\Gamma(\alpha_{ji})} \right] \tag{8}$$

where the first sum is over the combined states of its parents, and the second sum is over the conceivable states of F_j. Additionally, $N_{ji} = P\, k\, N_{jik}$ is the number of records needed to ensure that F_j is in its k-th state and that its parents are in their i-th configuration. The class variables have the same parents for all the graphs in the search space if we take into account a network linked to a particular class event since the links joining the class events are fixed. This suggests that the right-hand side's first sum in (4) is constant. Therefore, by taking only the features into account, the optimization in (9) can be accomplished. Any subset of C is a potential candidate for a feature's parents set, which simplifies the issue to m separate local optimizations. According to $G*$, F_j's parents are actually

$$\mathrm{C}_{F_j} = \arg^{\max}_{\mathrm{Pa}(F_j) \leq \mathrm{C}} \psi_a[F_j, \mathrm{Pa}(F_j)] \tag{9}$$

For each time $j = 1$, m. This is made possible by the bipartite separation of class events and features, but in most situations, directed cycles may be found in the graph that maximizes all local scores. Let k be the number of mixture components, X be the set of query variables, Z be the other variables (i.e., the number of values of C). Marginal distribution of X is evaluated by

adding values of C and Z from Equations (10)–(13):

$$P(X = x) = \sum_{c=1}^{i} \sum_{i} P(C = c, X = x, Z = z) \tag{10}$$

$$= \sum_{k=1}^{t} \sum_{t} P(c) \prod_{i=1}^{1} P(x_i|c) \prod_{j=1}^{A} P(z, |c) \tag{11}$$

$$= \sum_{i=1}^{k} P(c) \prod_{j=1}^{|e-1} P(x_j|c) \prod_{j=1}^{|\sum_j} P(= , |c) \tag{12}$$

$$= \sum_{c=1}^{t} P(c) \prod_{i=1}^{\|=1} P(x|c) \tag{13}$$

where the previous equality is true since, for any j, (j) 1 z_j $\sum$ P c z = . Therefore, it is simple to ignore the non-query variables Z while computing $P(X = x)$, and the computation of $P(X = x)$ takes $O(|X|k)$, regardless of $|Z|$. Bayesian network inference, in contrast, is worst-case exponential in |Z|. Conditional probabilities are effectively estimated as ratios of marginal probabilities $P(X = x|Y = y) = P(X = x, Y = y)/(Y = \dot{y})$. Mixture of trees, where each variable in each cluster is allowed to have one additional parent in addition to C, provides a slightly richer model than naive Bayes while still allowing for efficient inference.

The Bayesian Privacy (BP) is defined by Equation (14) for a protection mechanism M, a privacy leakage assault A, and a prior knowledge distribution F_0 concerning private data x.

$$\text{BP}(\mathcal{M}, \mathcal{A}, g, F_0) = \mathbf{KL}(F_0|F_{\mathcal{A}}(\bar{x}|\ \mathcal{M}(g(x)))) \tag{14}$$

where KL stands for the Kullback–Leibler divergence between the posterior distribution FA conditioned on disclosed information and the prior knowledge distribution F_0. This indicates that A and B are disjoint in set theory notation, i.e., $A \cap B = \varnothing$. $P(A|B) = P(A)$ or $P(B|A) = P(B)$, and $P(A\ B) = P(A)$ P. By using Equations (15) and (16), we can state the following:

$$P(A|B) = \frac{|A \cap B|}{|B|} = \frac{\frac{|A \cap B|}{|\cap|}|}{\frac{B|}{|M|}} = \frac{P(A \cap B)}{P(B)} \tag{15}$$

$$P(B|A) = \frac{|B \cap A|}{|A|} = \frac{\frac{|B \cap A|}{\left|\frac{1}{|c|}\right|}}{\underline{1i\pi}} = \frac{P(A \cap B)}{P(A)} \tag{16}$$

From Equations (17) and (18), it is immediately obvious that

$$P(A \cap B) = P(A|B)P(B) = P(B|A)P(A) \tag{17}$$

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \tag{18}$$

It is the Bayes theorem's most straightforward (and possibly most famous) statement. The generalized Bayes' formula is given by Equation (19) for each Ai.

$$P(A_i|B) = \frac{P(B|A_i)P(A_i)}{\sum_{j=1}^{n} P(B|A_j)P(A_j)} \tag{19}$$

It is given as Equation (20)

$$P(A|B) = \frac{P(B|A)P(A)}{P(B|A)P(A) + P(B|A^e)P(A^e)} \tag{20}$$

The total probability theorem dictates that Equations (19) and (20) follow from Equation (18) (Equations (15) and (16)). It is frequently believed that the data might emerge from two opposing hypotheses, H_1 and H_2, with $P(H_1) = 1\ P(H_2)$. It is also common to use the word 'model' for the word 'hypothesis.' Let D stand for the measured data. Then, Equation (21) provides posterior probability of hypothesis H_1.

$$P(H_1|D) = \frac{P(D|H_1)P(H_1)}{P(D|H_1)P(H_1) + P(D|H_2)P(H_2)} \tag{21}$$

and posterior probability of H_2 is represented by Equation (22)

$$P(H_2|D) = \frac{P(D|H_2)P(H_2)}{P(D|H_1)P(H_1) + P(D|H_2)P(H_2)} \tag{22}$$

From Equations (21) and (22), we obtain Equation (23)

$$\underbrace{\frac{P(H_1|D)}{P(H_2|D)}}_{\text{pesteriar odds}} = \underbrace{\frac{P(D|H_1)}{P(D|H_2)}}_{\text{Bayyes fartar } B_{12}} \cdot \underbrace{\frac{P(H_1)}{P(H_2)}}_{\text{prike olds}} \tag{23}$$

The ratio of H_1's posterior odds to its prior odds is known as Bayes factor. When compared to competing hypothesis H_2, Bayes factor is seen as a summary indicator of evidence data giving us supporting hypothesis H_1. Bayes factor and posterior odds are the same if the prior probabilities for H_1 and H_2 are the same. Let f_{XY} be joint PDF for continuous random variables X and Y. (x, y). Next, by Equations (24) and (25)

$$f_{X|Y}(x|y) = \frac{f_{XY}(x, y)}{f_Y(y)} \tag{24}$$

$$f_{Y|X}(y|x) = \frac{f_{XY}(x, y)}{f_X(x)} \tag{25}$$

so, that Equation (26) can be used to express Bayes' theorem for continuous variables

$$f_{X|Y}(x|y) = \frac{f_{Y|X}(y|x)f_X(x)}{f_Y(y)} \tag{26}$$

where $f_Y(y) = \int_X f_{Y|X}(y|x)f_X(x)dx = \int_{-\infty}^{+\infty} f_{XY}(x, y)dx$ because of the total probability theorem. These probabilities are used by the straightforward naive Bayes classifier to categorize an instance. By using Bayes' theorem (Equation (27)) and slightly

modifying the notation, we arrive at

$$P(y_j|\mathbf{x}_i) = \frac{P(\mathbf{x}_i|y_j)P(y_j)}{P(\mathbf{x}_i)} \tag{27}$$

As a result, the numerator can be recast as follows; for simplicity, we will skip the index I here and only use x by Equations (28)–(31):

$$P(\mathbf{x}|y_j)P(y_j) = P(\mathbf{x}, y_j) = P(x_1, x_2, \ldots, x_p, y_j) \tag{28}$$

$$= P(x_1|x_2, x_3, \ldots, x_p, y_j)P(x_2, x_3, \ldots, x_p, y_j) \text{ because } P(a, b) = P(a|b)P(b) \tag{29}$$

$$= P(x_1|x_2, x_3, \ldots, x_p, y_j)P(x_2|x_3, x_4, \ldots, x_p, y_j)P(x_3, x_4, \ldots, x_p, y_j) \tag{30}$$

$$= P(x_1|x_2, x_3, \ldots, x_p, y_j)P(x_2|x_3, x_4, \ldots, x_p, y_j)\cdots P(x_p|y_j)P(y_j) \tag{31}$$

Assume for the moment that each xi is distinct from the others. Strong and obviously broken in the majority of practical applications, this assumption is naïve, hence the term. Equations (32) and (33) are implied by this presumption

$$P(x_1|x_2, x_3, \ldots, x_p, y_j) = P(x_1|y_j) \tag{32}$$

$$P(y_j|\mathbf{x}) = \frac{\prod_{k=1}^{p} P(x_k|y_j)P(y_j)}{P(\mathbf{x})}(\psi_j).\mathrm{tr}(T) := \sum(\psi_i, T\psi_i) \tag{33}$$

Enter it into Equation (33) to get Equation (34)

$$P(\mathbf{x}|y_j)P(y_j) = P(x_1|y_j)\cdot \tag{34}$$

The denominator, $P(x)$, is independent of the class; it is the same for classes y_j and y_l, for instance. The scaling factor $P(x)$ makes sure that posterior probability $P(y_j|x)$ is appropriately scaled.

3.2. Back propagation with Kernelized support swarm machine (BPKSSM) based saline water analysis

Principal component analysis employing a kernel function is a technique for producing the conventional linear PCA in a high-dimensional space. KPCA performs nonlinear PCA in this feature space in a manner similar to performing nonlinear PCA in original input space.

Let H be a Hilbert space that implements the self-adjoin operator for the trace class G.

If and only if series $\sum_{\mathrm{set}}(\psi_i, |T|\psi_i\rangle$, with $|T| = \sqrt{T^*T}$ convergent for some ONB ψ_i then $T \in B(H)$ is considered to be a tumour trace class. By Equation (35) in this instance,

$$c_1 \parallel f \parallel^2 \leq \sum_{\mathrm{a}\in A} |\langle h_a, f\rangle|^2 \leq c_2 \parallel f \parallel^2 \quad \text{for all} f \in \mathcal{H} \tag{35}$$

Let$(h_\alpha)_{\alpha\in A}$ be a frame in H. Set $L{:}\mathcal{H} \to l^2$ from Equation (36)

$$L{:}f \to (\langle h_{\mathrm{a}}, f\rangle)_{\alpha\in A} \tag{36}$$

Then, $L^*{:}l^2 \to \mathcal{H}$ given by Equation (37),

$$L^*((c_\alpha)) = \sum_{\alpha\in A} c_\alpha h_\alpha \tag{37}$$

where $(c_a) \in l^2$; and $L^*L = \sum_{\alpha\in A} |h_\alpha\rangle\langle h_a|$

A function K: $S \times S \to C$ known as a positive definite (p.d.) kernel on S is one that, according to Equation (38)

$$\frac{1}{N}\sum_{i=1}^{N}\phi(\mathbf{x}_i) = 0. \tag{38}$$

for all$|\{x_i\}_{i=1}^{N} \subset S$, $\{c_i\}_{i=1}^{N} \subset$ C, and $N \in$ N

Given a p.d. kernel, a mapping : Φ: $S \to H$ (K) such that by Equation (1), RKHS H (K) and a p.d (39)

$$K(x, y) = \langle \Phi(x), \Phi(y) \rangle_{\mathcal{H}(K)} \tag{39}$$

A feature map is the term used for the function Φ in the problem. Additionally, the replicating property shown by Equation (40) holds:

$$f(x) = \langle K_x, f \rangle_{\mathcal{H}(K)} \tag{40}$$

for all $f \in H$ (K), and $x \in S$. H (K) may be selected as Hilbert completion of Equation (41)

$$\text{span}\{K_x := K(\cdot, x)\} \tag{41}$$

with respect to H (K)-inner product by Equation (42)

$$\langle \sum c_i K_{x_i}, \sum d_j K x_j \rangle_{\mu(K)} := \sum c_i d_j K(x_i, x_j) \tag{42}$$

In the new feature space, we can run normal PCA, although this can be quite expensive and ineffective. Fortunately, we can streamline the computation by using kernel approaches. First, using Equation (43), we assume that the expected new features have zero mean:

$$\frac{1}{N}\sum_{i=1}^{N}\phi(x_l) = 0 \tag{43}$$

The projected features' covariance matrix is $M \times M$, as determined by Equation (44)

$$\mathbf{C} = \frac{1}{N}\sum_{i=1}^{N}\phi(\mathbf{x}_i)\phi(\mathbf{x}_i)^{\mathrm{T}}, \tag{44}$$

Equation (45) provides its eigenvalues and eigenvectors

$$\mathbf{C}\mathbf{v}_k = \lambda_k \mathbf{v}_k \tag{45}$$

where $k = 1, 2, \cdots, M$. Following Mercer's theory for kernels, we know that kernel κ takes the form by Equation (46)

$$\kappa(\mathbf{x_i}, \mathbf{x_j}) = \phi(x_i)^T \phi(x_j) \tag{46}$$

Substitute Equation (46) into Equation (45) and multiply both sides of Equation (47) by φ(xl)

$$\frac{1}{N}\kappa(x_l, x_i)\sum_{i=1}^{N} a_{kj}\kappa(x_i, x_j) = \lambda_k \sum_{i=1}^{N}\kappa(x_l, x_i) \tag{47}$$

which can be re-written as Equations (48)–(50)

$$\frac{1}{N}\kappa(x_l, x_i)\sum_{i=1}^{N} a_{kj}\kappa(x_i, x_j) = \lambda_k \sum_{i=1}^{N}\kappa(x_l, x_i) \tag{48}$$

$$\phi(x_i)\sum_{j=1}^{N} a_{kj}\phi(x_i)\phi(x_j) = \mathbf{K}^2\mathbf{a_k} = \lambda_k N\mathbf{a_k} \tag{49}$$

$$\mathbf{K} = \kappa(x_i, x_j) \tag{50}$$

a_k is the eigenvector which is an N dimensional column vector of a_{ki}. a_k can be solved by Equation (51)

$$\mathbf{K}\mathbf{a_k} = \lambda_k N\mathbf{a_k} \tag{51}$$

The resulting kernel principal components transformation is given by Equation (52)

$$\hat{x} = \phi(\mathbf{x})^T\mathbf{u_k} = \sum_{i=1}^{N} a_{ki}\kappa(\mathbf{x}, \mathbf{x}_i) \tag{52}$$

Given an uncentered kernel matrix, we compute the zero mean of the kernel, thus by Equations (53) and (54)

$$\hat{\mathbf{K}} = \|\phi(x_i) - \frac{1}{N}\sum_{j=1}^{N}\phi(x_j)\|_2 = \left(\phi(x_i) - \frac{1}{N}\sum_{j=1}^{N}\phi(x_j)\right)^T \tag{53}$$

$$\left(\phi(x_i) - \frac{1}{N}\sum_{j=1}^{N}\phi(x_j)\right) \tag{54}$$

After expansion, we have that from Equation (55)

$$= K_{ij} - \sum_{i=1}^{N}\phi(x_i)^T\phi(x_j)\frac{1}{N} - \frac{1}{N}\sum_{i=1}^{N}\phi(x_j)^T\phi(x_i) - \frac{1}{N}\sum_{i=1}^{N}\phi(x_j)\phi(x_j) \tag{55}$$

This can be rewritten in short as Equation (56)

$$\hat{\mathbf{K}} = \mathbf{K} - 1_{1/N}\mathbf{K} - \mathbf{K}1_{1/N} + 1_{1/N}\mathbf{K}1_{1/N} \tag{56}$$

where $\mathbf{K} = \mathbf{K}_{ij}.\hat{\mathbf{K}}$ is called the Gram matrix (normalized kernel matrix).

A shallow FFMLP_ANN with one hidden layer is used to replicate the 1D extended Nernst-Planck mechanistic ion transport method (pEnPEn) from the previous section. In hidden and output layers, a hyperbolic tangent transfer function is used. Figure 3 shows the ANN's structural layout. Layer charge X, layer thickness x, trans membrane velocity v, and salt supply concentration c_j are all inputs to ANN. Ionic retention of ion j R_j is output to ANN.

4. PERFORMANCE ANALYSIS

The experiment was carried out using a computer with the following technological specs: Intel Core i5 7200U, 8 GB of RAM, a 1 TB hard drive, and NVIDIA GTX 760MX graphics. Python 3.5 environments were used to mimic the implementation of the suggested approach. We conducted a statistical study by evaluating predicted performance to establish the findings of the suggested strategy.

Dataset description: The feed flow rate ($F = 400$–600 L/h), permeate flux (P_{flux} (L/h·m^2)), condenser inlet temperature ($T_{\text{cond}} = 20$–30 °C), evaporator inlet temperature ($T_{\text{evap}} = 60$–80 °C), and feed salt concentration ($S = 35$–140 g/L) made up

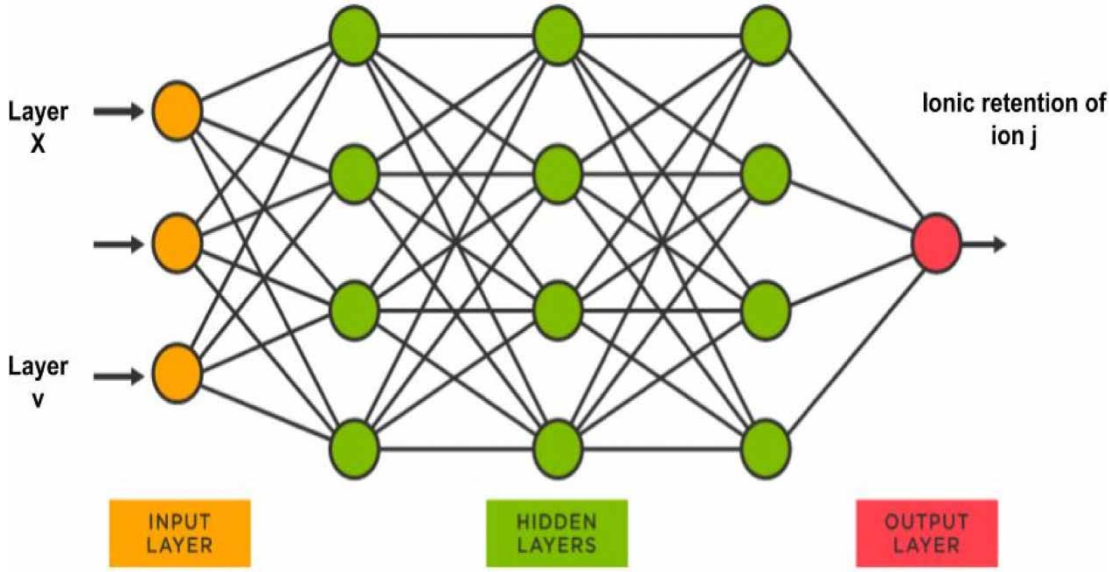


Figure 3 | Architecture of BPKSSM.

the datasets. The principal output was permeated flux. In order to train NN, data have also been separated into three categories: training, validation, and testing. Training division is typically utilized to get model specifications. While testing division confirms its performance, validation division checks precision of continuous training to prevent over fitting.

Table 1 gives a comparative analysis between proposed and existing technique based on salt water analysis. Here the salt water analysis has been carried out based on salinity and temperature analysis of water like S = 35–140 g/L, T_{cond} = 20–30 °C, T_{evap} = 60–80 °C. Parameters analysed are in terms of accuracy, precision, recall, specificity, computational cost, and Kappa coefficient.

Figure 4(a)–4(f) gives comparative analysis between the proposed and existing technique for S = 35–140 g/L based on water salinity. The proposed technique attained accuracy of 85%, precision of 75%, recall of 75%, specificity of 75%, computational cost of 51%, Kappa coefficient of 68%, existing SUTRA attained accuracy of 81%, precision of 72%, recall of 68%, specificity of 71%, computational cost of 45%, Kappa coefficient of 61%, ANN attained accuracy of 83%, precision of 74%, recall of 72%, specificity of 73%, computational cost of 48%, and Kappa coefficient of 63%.

Figure 5(a)–5(f) gives a comparative analysis between proposed and existing techniques for T_{cond} = 20–30 °C based on water salinity. The proposed technique attained accuracy of 86%, precision of 81%, recall of 74%, specificity of 78%, computational cost of 53%, Kappa coefficient of 73%, existing SUTRA attained accuracy of 82%, precision of 73%, recall of 70%, specificity of 72%, computational cost of 48%, Kappa coefficient of 68%, ANN attained accuracy of 84%, precision of 78%, recall of 72%, specificity of 75%, computational cost of 52%, and Kappa coefficient of 71%.

Table 1 | Comparative analysis between proposed and existing techniques based on various saline water analyses

Salt water analysis	Techniques	Accuracy	Precision	Recall	Specificity	Computational cost	Kappa coefficient
S = 35–140 g/L	SUTRA	81	72	68	71	45	61
	ANN	83	74	72	73	48	63
	OMTP_SWL_MLA	85	75	75	75	51	65
T_{cond} = 20–30 °C	SUTRA	82	73	70	72	48	68
	ANN	84	78	72	75	52	71
	OMTP_SWL_MLA	86	81	74	78	53	73
T_{evap} = 60–80 °C	SUTRA	85	75	74	75	49	71
	ANN	88	79	76	79	55	75
	OMTP_SWL_MLA	92	83	78	81	59	78

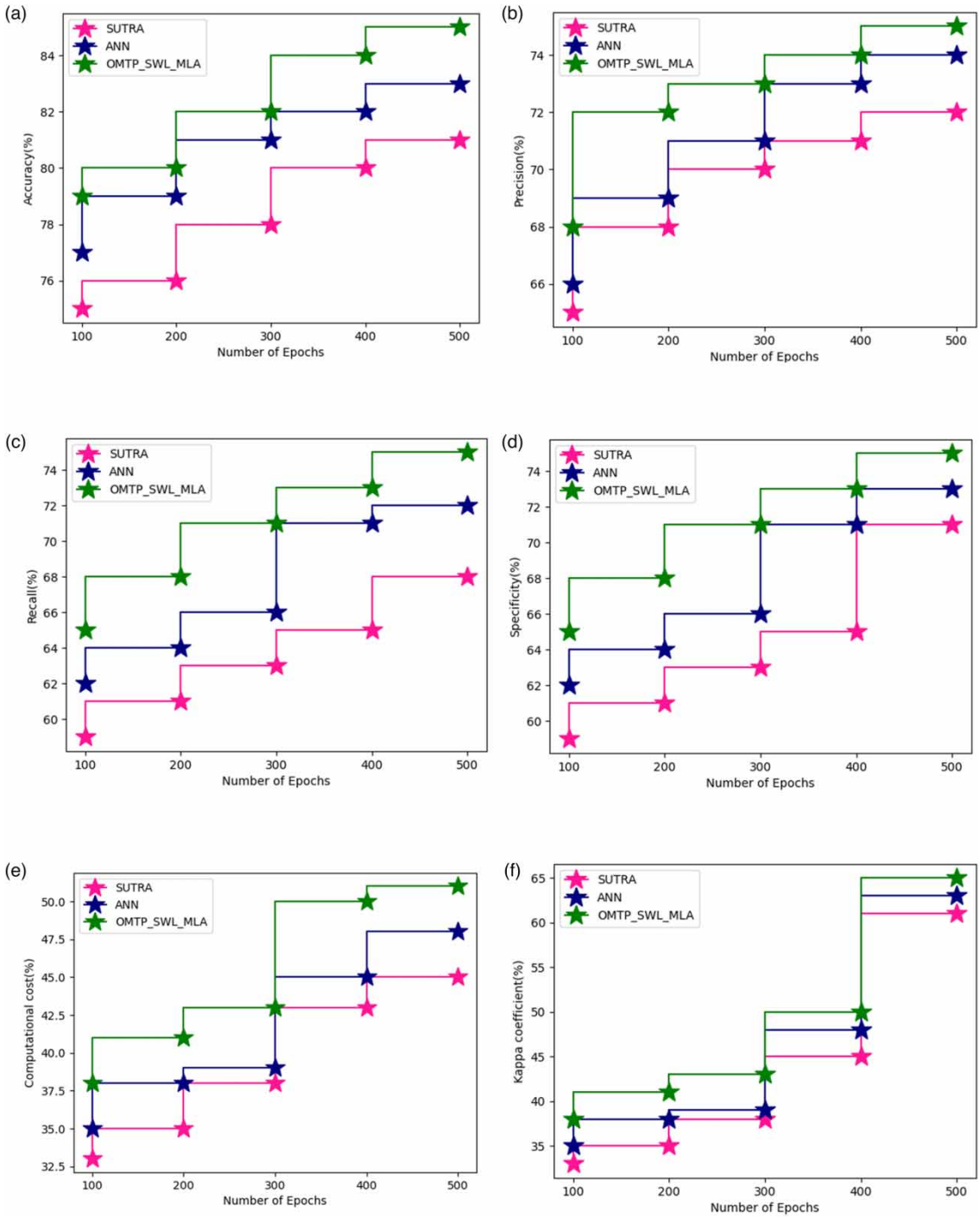


Figure 4 | Comparative analysis between proposed and existing technique for $S = 35–140$ g/L based on water salinity in terms of (a) Accuracy, (b) Precision, (c) Recall, (d) Specificity, (e) Computational cost, (f) Kappa coefficient.

Figure 6(a)–6(f) gives a comparative analysis between proposed and existing techniques for $T_{evap} = 60–80$ °C based on water salinity. The proposed technique attained accuracy of 92%, precision of 83%, recall of 78%, specificity of 81%, computational cost of 59%, Kappa coefficient of 78%, existing SUTRA attained accuracy of 85%, precision of 75%, recall of 74%, specificity of 75%, computational cost of 49%, Kappa coefficient of 71%, ANN attained accuracy of 88%, precision of 79%, recall of 76%, specificity of 79%, computational cost of 55%, and Kappa coefficient of 75%.

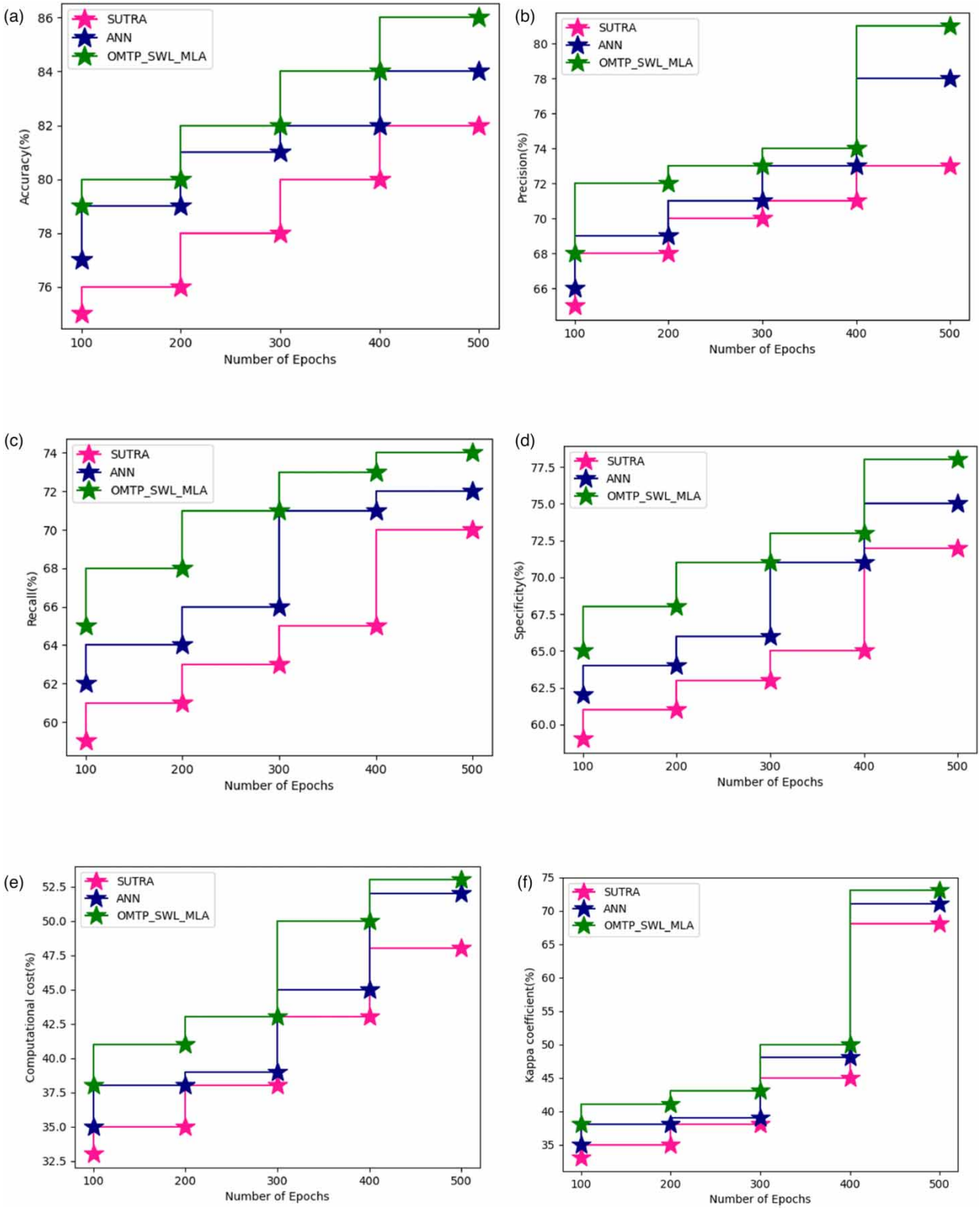


Figure 5 | Comparative analysis between proposed and existing techniques for $T_{cond} = 20–30$ °C based on water salinity in terms of (a) Accuracy, (b) Precision, (c) Recall, (d) Specificity, (e) Computational cost, (f) Kappa coefficient.

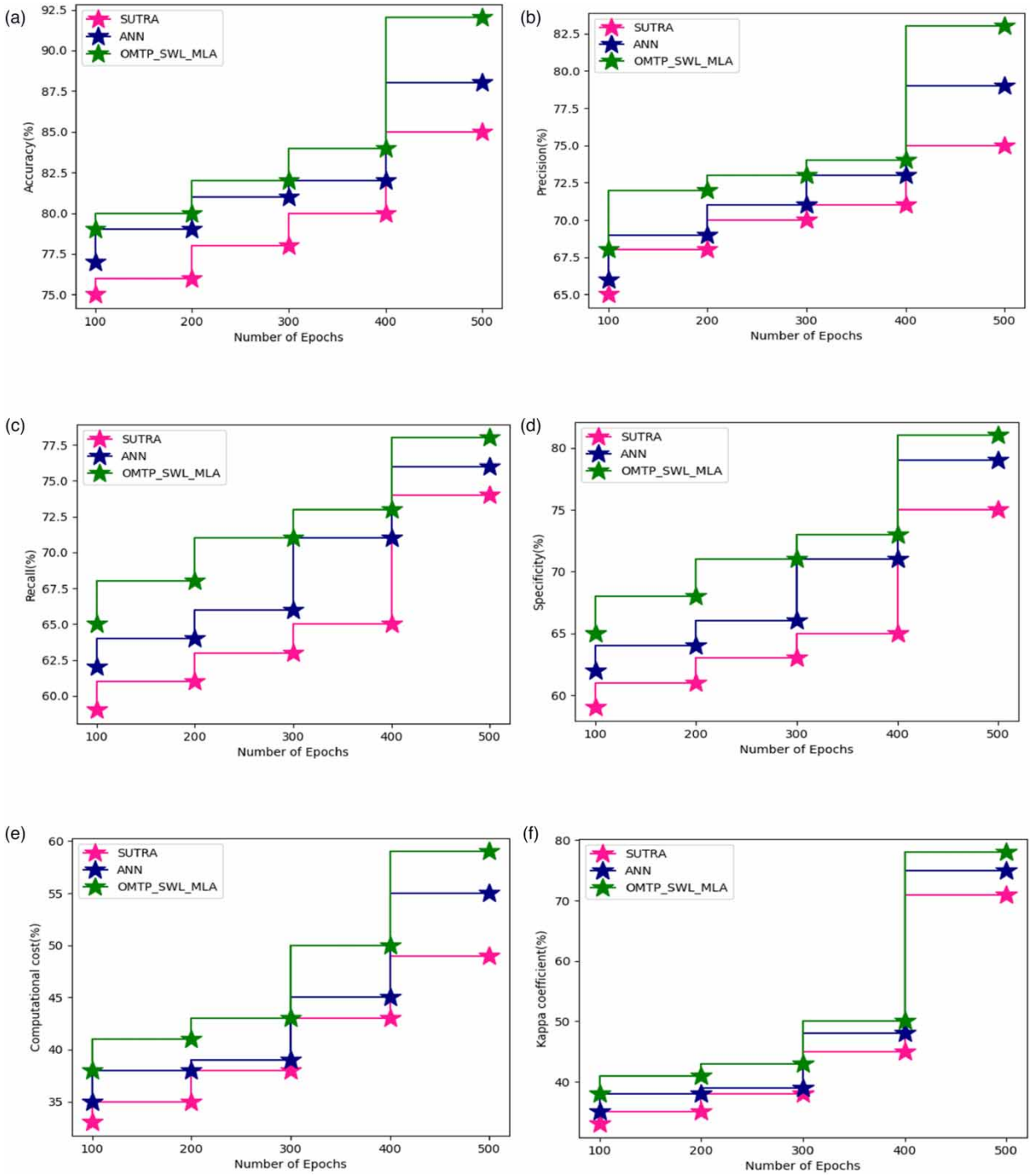


Figure 6 | Comparative analysis between proposed and existing techniques for $T_{evap} = 60–80$ °C based on water salinity in terms of (a) Accuracy, (b) Precision, (c) Recall, (d) Specificity, (e) Computational cost, (f) Kappa coefficient.

5. DISCUSSION

Figure 4(a)–4(f) shows that the majority of samples were correctly classified by our classification model, while only a tiny fraction of samples was wrongly classified. Additionally, Figure 5(a)–5(f) showed that if there are more training data available for the method to be trained, the overall accuracy of method may rise. The model's specificity metrics demonstrated a true

negative ratio of 0.89, indicating that it correctly identified the majority of the negative predictions. On the other hand, the model correctly predicted 92% of the positive test data outcomes. Additionally, Figure 5(a)–5(f) shows how the model demonstrated a specificity score, which made the model efficient and resilient by classifying more positive predictions over wrong misclassification. The proposed method maintained between precision and recall, according to an f-score. Figure 6(a)–6(f) displays the individual classification scores for the target classes in order to demonstrate cutting-edge findings and offer a full analysis report for assessing the suggested strategy.

6. CONCLUSION

This research proposed a novel method for optimization, as well as modelling of the treatment process using ML techniques using molecular separation-based reverse osmosis. Bayesian optimization and water saline particle analysis have been carried out using back propagation with Kernelized support swarm machine. The proposed models were evaluated and examined by some statistical parameters where with the training data's constrained range of class labels, the system was unable to categorize salts with salinities greater than 100 ppt. The proposed method attained accuracy of 92%, precision of 83%, recall of 78%, specificity of 81%, computational cost of 59%, and Kappa coefficient of 78%. Further, we intend to examine the applicability of DL methods in water treatment techniques by integrating our suggested approach with contemporary water treatment facilities in further research.

DATA AVAILABILITY STATEMENT

All relevant data are included in the paper or its Supplementary Information.

CONFLICT OF INTEREST

The authors declare there is no conflict.

REFERENCES

Aghilesh, K., Mungray, A., Agarwal, S., Ali, J. & Garg, M. C. 2021 Performance optimisation of forward-osmosis membrane system using machine learning for the treatment of textile industry wastewater. *Journal of Cleaner Production* **289**, 125690.

Alshehri, M., Kumar, M., Bhardwaj, A., Mishra, S. & Gyani, J. 2021 Deep learning based approach to classify saline particles in sea water. *Water* **13** (9), 1251.

Batchuluun, S., Matsune, H., Shiomori, K., Bayanjargal, O. & Baasankhuu, T. 2021 Preparation of polystyrene microcapsules containing saline water droplets via solvent evaporation method and their structural distribution analysis by machine learning. *Journal of Chemical Engineering of Japan* **54** (9), 517–524.

Bonny, T., Kashkash, M. & Ahmed, F. 2022 An efficient deep reinforcement machine learning-based control reverse osmosis system for water desalination. *Desalination* **522**, 115443.

Dargam, F., Perz, E., Bergmann, S., Rodionova, E., Sousa, P., Souza, F. A. A. & Bonachela, P. Z. 2020 Supporting operational decisions on desalination plants from process modelling and simulation to monitoring and automated control with machine learning. In: *International Conference on Decision Support System Technology* (J. M. Moreno-Jiménez, I. Linden, F. Dargam & Uchitha Jayawickrama, eds.). Springer, Cham, pp. 150–164.

Doña, C., Chang, N. B., Caselles, V., Sánchez, J. M., Pérez-Planells, L., Bisquert, M. D. M., García-Santos, V., Imen, S. & Camacho, A. 2016 Monitoring hydrological patterns of temporary lakes using remote sensing and machine learning models: case study of la Mancha Húmeda Biosphere Reserve in central Spain. *Remote Sensing* **8** (8), 618.

Elsheikh, A. H., Shanmugan, S., Sathyamurthy, R., Thakur, A. K., Issa, M., Panchal, H. & Sharifpur, M. 2022 Low-cost bilayered structure for improving the performance of solar stills: performance/cost analysis and water yield prediction using machine learning. *Sustainable Energy Technologies and Assessments* **49**, 101783.

Feizizadeh, B., Garajeh, M. K., Lakes, T. & Blaschke, T. 2021 A deep learning convolutional neural network algorithm for detecting saline flow sources and mapping the environmental impacts of the Urmia Lake drought in Iran. *Catena* **207**, 105585.

Jiang, W., Pokharel, B., Lin, L., Cao, H., Carroll, K. C., Zhang, Y. & Xu, P. 2021 Analysis and prediction of produced water quantity and quality in the Permian Basin using machine learning techniques. *Science of The Total Environment* **801**, 149693.

Lee, W. H., Park, C. Y., Diaz, D., Rodriguez, K. L., Chung, J., Church, J., Willner, M. R., Lundin, J. G. & Paynter, D. M. 2022 Predicting bilgewater emulsion stability by oil separation using image processing and machine learning. *Water Research* **223**, 118977.

Li, Y., Wang, X., Zhao, Z., Han, S. & Liu, Z. 2020 Lagoon water quality monitoring based on digital image analysis and machine learning estimators. *Water Research* **172**, 115471.

Milosavljević, A. 2020 Identification of salt deposits on seismic images using deep learning method for semantic segmentation. *ISPRS International Journal of Geo-Information* **9** (1), 24.

Odabaşı, Ç., Dologlu, P., Gülmez, F., Kuşoğlu, G. & Çağlar, Ö. 2022 Investigation of the factors affecting reverse osmosis membrane performance using machine-learning techniques. *Computers & Chemical Engineering* **159**, 107669.

Rall, D., Schweidtmann, A. M., Kruse, M., Evdochenko, E., Mitsos, A. & Wessling, M. 2020 Multi-scale membrane process optimization with high-fidelity ion transport models through machine learning. *Journal of Membrane Science* **608**, 118208.

Sagastibeltza, N., Salazar-Ramirez, A., Yera, A., Martinez, R., Muguerza, J., Sanchez, N. C. & Gil, M. A. A. 2022 Preliminary study on the detection of autonomic dysreflexia using machine learning techniques. In: *International Conference on Computer Science, Electronics and Industrial Engineering (CSEI)* (M. V. Garcia, F. Fernández-Peña & C. Gordón-Gallegos, eds.). Springer, Cham, pp. 341–351.

Sahour, H., Gholami, V. & Vazifedan, M. 2020 A comparative analysis of statistical and machine learning techniques for mapping the spatial distribution of groundwater salinity in a coastal aquifer. *Journal of Hydrology* **591**, 125321.

Son, M., Yoon, N., Jeong, K., Abass, A., Logan, B. E. & Cho, K. H. 2021 Deep learning for pH prediction in water desalination using membrane capacitive deionization. *Desalination* **516**, 115233.

Viet, N. D., Im, S. J., Kim, C. M. & Jang, A. 2021 An osmotic membrane bioreactor–clarifier system with a deep learning model for simultaneous reduction of salt accumulation and membrane fouling. *Chemosphere* **272**, 129872.

Wang, H., Zhang, X., Li, P., Sun, J., Yan, P., Zhang, X. & Liu, Y. 2020 A new approach for unqualified salted sea cucumber identification: integration of image texture and machine learning under the pressure contact. *Journal of Sensors* **2020** (4), 1–13.

Zhang, Y., Thangavelu, L., Taban, T. Z., Abdelbasset, W. K., Suksatan, W., Sarjadi, M. S., Rahman, M. L., Sarkar, S. M., Alashwal, M., Zwawi, M. & Algarni, M. 2022 Development of hybrid machine learning model for simulation of chemical reactors in water treatment applications: absorption in amino acid. *Environmental Technology & Innovation* **27**, 102417.

First received 19 October 2022; accepted in revised form 11 December 2022. Available online 29 December 2022