

Article

Improvement of Cyber-Attack Detection Accuracy from Urban Water Systems Using Extreme Learning Machine

Young Hwan Choi ¹, Ali Sadollah ²  and Joong Hoon Kim ^{3,*} ¹ Department of Civil Engineering, Gyeongnam National University of Science and Technology, Jinju 52725, Korea; yh.choi@gntech.ac.kr² Department of Mechanical Engineering, University of Science and Culture, Tehran 1461968151, Iran; sadollah@usc.ac.ir³ School of Civil, Environmental and Architectural Engineering, Korea University, Seoul 02841, Korea

* Correspondence: jaykim@korea.ac.kr

Received: 22 September 2020; Accepted: 16 November 2020; Published: 18 November 2020



Abstract: This study proposes a novel detection model for the detection of cyber-attacks using remote sensing data on water distribution systems (i.e., pipe flow sensor, nodal pressure sensor, tank water level sensor, and programmable logic controllers) by machine learning approaches. The most commonly used and well-known machine learning algorithms (i.e., k-nearest neighbor, support vector machine, artificial neural network, and extreme learning machine) were compared to determine the one with the best detection performance. After identifying the best algorithm, several improved versions of the algorithm are compared and analyzed according to their characteristics. Their quantitative performances and abilities to correctly classify the state of the urban water system under cyber-attack were measured using various performance indices. Among the algorithms tested, the extreme learning machine (ELM) was found to exhibit the best performance. Moreover, this study not only has identified excellent algorithm among the compared algorithms but also has considered an improved version of the outstanding algorithm. Furthermore, the comparison was performed using various representative performance indices to quantitatively measure the prediction accuracy and select the most appropriate model. Therefore, this study provides a new perspective on the characteristics of various versions of machine learning algorithms and their application to different problems, and this study may be referenced as a case study for future cyber-attack detection fields.

Keywords: water distribution systems; cyber-attack detection; remote sensing data and controller; extreme learning machine; machine learning algorithms

1. Introduction

Most infrastructure systems (e.g., water distribution systems (WDSs), power grid systems, and telecommunication systems) have been monitoring and controlling their remote sensing data on supervisory control and data acquisition (SCADA) systems and programmable logic controllers (PLCs) to efficiently operate and manage. These supervisory and logical systems can be described as cyber-physical systems (CPSs) that are connected by the cyber network to physical devices. Because a CPS can monitor and control the physical processes of equipment in real-time, it is practical and convenient to make use of such systems in the operation of infrastructure systems, wherein even a minor malfunction can cause serious damage.

In the past few years, the CPSs have been used in several infrastructure systems from power grid systems to telecommunications systems. They have also been used in water utilities, such as in WDSs,

water supply reservoirs, and water treatment systems [1–3]. Although the utilization of CPSs in these infrastructure systems has proven to be efficient, they are vulnerable to cyber–physical attacks. The PLCs, actuators, and sensors that the CPSs make use of may be compromised by information disturbance to the sensor and communication server. Recently, major infrastructure systems (e.g., power grid, communication, and water networks) have been damaged by cyber-attacks; hence, prevention against cyber-attacks has been a focus in engineering fields, particularly for WDSs [3–5]. WDSs can directly affect public health and the environment. If the WDSs are damaged by cyber-attacks, there may be serious consequences, such as water shortages or degrading water quality [6].

Recently, WDSs technologies have been developed to use SCADA systems and advanced metering infrastructures (AMIs). These technologies are used to forecast the future state of the systems, such as the values of demand, pressure, and water quality, to detect abnormal conditions in the system (e.g., pipe breakage, leakage, and contamination events) [7–10] and to determine the optimal operating conditions for pumps and valves [11,12]. With these developments comes the increased risk of cyber-attacks on the WDSs. Therefore, emphasis must be placed not only on the development of WDSs using the latest technology, but also on the development of technology to minimize the damage brought about by cyber-attacks and detect such attacks in real-time situations. Previous studies conducted dealt primarily with cyber-attacks on power grid systems. Initially, the studies focused on the missing data collecting for the communication problems and the malfunction of measurement error between storage sources and sensors in SCADA systems [13,14]. The focus of later studies shifted to the interactions between PLCs and actuators, such as scenarios wherein incorrect information is deliberately inputted, leading to system overload and malfunctions [15–17].

Recent studies performed real-time state estimations of cyber-attacks. Given the ability of modern technology to generate and store a large amount of data, real-time data forecasting is an efficient approach for optimal and reliable system operations. In the case of power grid systems, this process involves estimating unknown data based on the real-time measurements of system variables, such as flow and voltage values [15,18–21]. To detect false data cyber-attacks against the state of the systems, studies have undertaken a computationally efficient heuristic and Bayesian approach [20]. Liu et al. [17] proposed a framework for identifying a set of power lines that could be overloaded due to false data inputs, and should be strictly monitored to ensure the reliability and safety of power systems. However, given the recent attacks and minute malfunctions in the system, models that more accurately detect such events are essential. In response to this, recent studies have applied machine learning approaches to the development of such models [22–24].

On the other hand, with regard to WDSs, which are closely related to public health and environment, few attempts have been made to solve the problem of detecting cyber-attacks on water network systems. Amin et al. [25] proposed a hydrodynamic model for the detection of cyber-attacks and the isolation of the unpressured roof network in an irrigation field. Perelman and Amin [26] developed a model for simulating attacks on WDSs considering hydraulic balance and energy consumption. Their proposed model assessed the vulnerability of the systems under cyber-attack conditions by applying an attack situation (e.g., abnormal condition), such as removing one pipe. However, their suggested model could not consider various attack scenarios.

With improvements in the handling technology of big-data, studies have then focused on the clustering and classification of such data using artificial intelligence (AI) techniques, such as machine learning. Almalawi et al. [27] introduced the cyber-attack detection approach based on a data-driven clustering technique in the SCADA system of a simple WDS model. Taormina et al. [28] developed the EPANET-CPA (EPANET based Cyber physical attack) based on hydraulic engine (i.e., EPANET 2.0 [29]), which can simulate various cyber-attack scenarios for hydraulic and water quality failures, particularly focusing on CPS components (e.g., sensors, PLCs, and SCADAs). Tanomina and Galelli [30] studied the detection of cyber-attacks using deep-learning techniques; however, their model focused on simulating and characterizing cyber-attacks rather than their detection and identification. The aforementioned studies demonstrate that there is a recent trend in the use of

machine learning to accurately detect cyber-attacks in the WDS field. However, since the detection performance of cyber-attack differs depending on the kinds of machine learning algorithms used, a study that determines the most suitable machine learning approach for the detection of cyber-attacks through the quantitative comparison of the performances of various machine learning algorithms will be needed.

In summary of the above statements, in the past, the CPSs have been used in several infrastructure systems such as power grid systems and telecommunications systems. However, recent WDSs fields have also used CPSs. For this reason, many researchers developed the operation and management approach of CPSs and cyber-attack detection and prevention approaches, because it can be easy to access in a cyber network.

Therefore, this study proposes a novel detection model coupled with machine learning (ML) approaches for the effective detection of cyber-attacks on WDSs. For this reason, this study applied and compared several types of machine learning approaches (i.e., extreme learning machine (ELM), k-nearest neighbor (KNN) algorithm, Support Vector Machine (SVM), and artificial neural networks (ANNs)) for finding the best ML algorithm. This study not only searches for the best performing algorithm through simple algorithm comparison but also considers the improved version of the basic algorithm to show the best performance. Therefore, this study could become a foundational study in the future development of cyber-attack detection techniques in WDSs fields.

Furthermore, this study used a dataset from the Battle of the Attack Detection Algorithms (BATADAL [31]) for the various cyber-attack scenarios and a comparison was performed using various representative performance indices which measure the prediction accuracy quantitatively and select the most appropriate algorithm.

2. Cyber-Attack Situations in WDSs

2.1. Security Goals and Cyber-Physical Attacks

The goal of WDSs is to supply sufficient clean water to meet the needs of the population. For this reason, highly complex water systems should be monitored and controlled in a timely manner to achieve high system efficiency and reliable water supply. Recently, water infrastructure systems have been making use of CPSs for automatic and accurate system operation based on SCADA systems, sensors, actuators, and communication systems.

With water infrastructure systems that are more connected to telecommunication systems, they are also more vulnerable to cyber-attacks. Therefore, efforts should be made to improve the detection and identification of cyber-attacks on SCADA systems and physical water systems. Thus, this study proposes an effective detection model based on various machine learning approaches. The BATADAL dataset (<http://batadal.net/data.html>) was used with the specific details of the attack scenarios demonstrated in the following sections.

2.2. Attack Model

The datasets used in this study are the only open-source datasets that consider the cyber-attack conditions of the WDSs. The cyber-attack data introduced in the World Environmental and Water Resources Congress held in Sacramento, California [28] is discussed in detail in the following paragraphs. The data generated on the potential effect of a cyber-attack on C-town's WDSs (see Figure 1) are utilized for the battle of the WDSs [28]. The C-town network consists of 429 pipes, 388 nodes, 7 tanks, 11 pumps, 4 valves, and a reservoir. Because the attacks are conducted through the malicious control of actuators, modification of the PLC control settings, or delivery of erroneous information due to malfunctions in the communication systems, the components' (e.g., pumps, valves and tank) sensors are connected to 9 PLCs and are located near the components. The training input data covered the SCADA values of 43 system components (i.e., the water level of 7 tanks, the status and water flow of 11 pumps, and

the status and water flow of a valve and the pressure of 12 junctions) and the training output data reflected the attack or non-attack conditions.

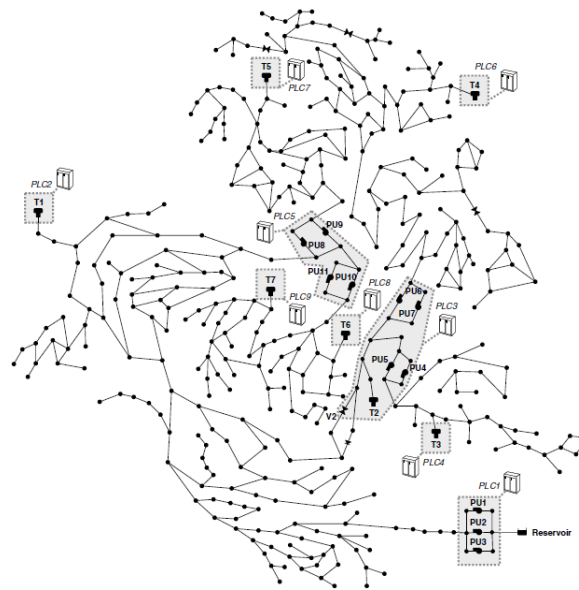


Figure 1. Description of C-town WDSs (water distribution systems) (Note: T: tank, V: valve, PU: pump, PLC: programming logical controller).

2.3. Attack Scenarios and Specifications

For the BATADAL dataset, 6 attack scenarios were considered. The attack scenarios were limited to hydraulic problems, such as disruption of pump operations, tank overflow, or depletion. The attack scenarios are distributed across the whole of the simulation period, therefore a total of 383 time steps were set as “under-attack” conditions, whereas the other 12,555 data were denoted as “non-attack” conditions. The test dataset had a length of 2089 data with 407 time steps labeled as “under-attack” conditions.

To apply the attack scenarios from the normal condition, the tank’s water level was considered to be 0.25 m (T1 and T2), 0.75 m (T5), 1 m (T3, T4, and T7), 2 m (T6) as an initial condition; this individual simulation was used for showing the characteristic of the effect for cyber–physical attacks [28]. The specifications of each attack scenario are shown in Table 1, and Scenario 1, 2, 3, and 5 are set to perform repeated generation during the simulation period because the malfunction of the tank or pump can be often generated as they are connected to several sensors or actuators.

- **Scenario 1:** This scenario focuses on a direct attack on the components of WDSs, such as pumps, valves, and tanks. For example, a tank overflow may be caused by a direct attack on a pumping station through the activation of unscheduled pumps.
- **Scenario 2:** This scenario is caused by a disturbance in the reading or transportation of data between the sensors and PLC. For instance, manipulated water level readings can lead to the depletion or low levels of water in the tanks. This problem may be due to the physical manipulation of the water level sensor or the miscommunication of information between the sensor and PLC.
- **Scenario 3:** This scenario relates to the connection among PLCs. If Tank 1 is connected to PLC 1, and the information of Tank 1 (e.g., water level) is intercepted by the attack and is transmitted instead to PLC 2, the other tank connected to PLC 2 will either overflow or be depleted.
- **Scenario 4:** This scenario is an attack to conceal the PLC connection problem (Scenario 3) and disrupt the information between SCADA and PLC. An example would be a situation wherein there is a problem in the identification of attacks due to the malfunction of communication links between the PLC and SCADA.

- **Scenario 5:** This scenario focuses on the SCADA data transportation problem. This attack scenario entails the alteration of the packages being sent by SCADA to change the operations of a PLC it supervises. In particular, the communication link between SCADA and PLC is attacked, resulting in pump activation and tank overflow.
- **Scenario 6:** This scenario contains random multiple combinations of attacks on PLCs.

Table 1. Attack specifications in training and test input data.

Attack Cases	Scenarios	Duration (hr)	Period (YYYY.MM.DD HH: mm)	Attack Descriptions
1	Scenario 2	50	2016.09.13 23:00–2016.09.16. 16:00	Attacker changes L_T7 thresholds controlling PU10 and PU11 by altering SCADA transmission to PLC5. This causes low levels in T7.
2	Scenario 2	24	2016.09.26 11:00–2016.09.27. 10:00	Attacker changes L_T7 thresholds controlling PU10 and PU11 by altering SCADA transmission to PLC5. This causes low levels in T7.
3	Scenario 3	60	2016.10.09 09:00–2016.10.11. 20:00	Attacker alters L_T1 readings arriving to PLC2 with a constant low level. PLC1 receives the manipulated readings from PLC2 and keeps Pumps PU1 and PU2 on, driving T1 to overflow.
4	Scenario 3	94	2016.10.29 19:00–2016.11.02. 16:00	Attacker alters L_T1 readings arriving to PLC2 with a constant low level. PLC1 receives the manipulated readings from PLC2 and keeps Pumps PU1 and PU2 on, driving T1 to overflow.
5	Scenario 1	60	2016.11.26 17:00–2016.11.29. 04:00	Working speed of PU7 reduced to 0.9 of nominal speed causes lower water levels in T4.
6	Scenario 1	94	2016.12.06 07:00–2016.12.10. 04:00	Working speed of PU7 reduced to 0.7 of nominal speed causes lower water levels in T4.
7	Scenario 1	110	2016.12.14 15:00–2016.12.19. 04:00	Working speed of PU7 reduced to 0.7 of nominal speed causes lower water levels in T4.
8	Scenario 1	70	2017.01.16 09:00–2017.01.19. 06:00	Working speed of PU7 reduced to 0.7 of nominal speed causes lower water levels in T4.
9	Scenario 3	65	2017.01.30 08:00–2017.02.02. 17:00	Attacker alters L_T2 readings arriving to PLC3, which reads a constant low level and forces Valve V2 open, leading T2 to overflow.
10	Scenario 5	31	2017.02.09 03:00–2017.02.10. 09:00	Malicious activation of Pump PU3.
11	Scenario 5	31	2017.02.12 01:00–2017.02.17. 07:00	Malicious activation of Pump PU3.
12	Scenario 3	100	2017.02.24 05:00–2017.02.28. 08:00	Attacker alters L_T2 readings arriving to PLC3, which reads a constant low level and forces Valve V2 open, leading T2 to overflow.
13	Scenario 6	80	2017.03.10 14:00–2017.03.13. 21:00	Attacker changes L_T7 thresholds controlling PU10 and PU11 by gaining control of PLC5, causing the pumps to switch continuously.
14	Scenario 4	30	2017.03.25 20:00–2017.03.27. 01:00	Alteration of T4 signal arriving at PLC6.

3. Application and Comparison of Various Classification Methods

The proposed novel detection model is a tool for the effective detection of cyber-attacks on WDSs. To develop detection approaches, this study applied a set of well-known and commonly used machine learning approaches. The applied cyber-attacks scenarios are expected to lead to anomalous hydraulic results in various WDSs components (e.g., the nodal pressure, tank level, and pipe flow), and to a corresponding significant system error.

Therefore, each machine learning approach was trained using historical SCADA data representing normal operating conditions. Then, these approaches were simulated with new SCADA data, and the reconstruction error produced at each time step was monitored. The monitoring data of the WDSs is classified as under attack if the average reconstruction error across all system variables is larger than a user-defined threshold (this study used 0.95 [30]) and is classified as safe otherwise.

The prediction performance of each algorithm is evaluated using the performance indices that have various characteristics and the most suitable algorithm for the detection model was identified by comparing the performances of the top four standard algorithms (i.e., KNN algorithm, SVM, ANNs, and ELM). Then, the classification results of the model using improved versions of the best algorithm were compared and analyzed in detail.

3.1. K-Nearest Neighbor Algorithm

The K-Nearest Neighbor (KNN) strategy is one of the simplest and most fundamental of the classification approaches. This method is useful for classification where there is little to no prior information on the distribution of data [32,33]. Table 2 presents the pseudo code of KNN [34].

Table 2. Pseudocode of the k-Nearest neighbor algorithm.

Definition: X : training data, Y : class labels of X , x : unknown sample
Initialize: Set variables X, Y, x
for $i = 1$ to m
Compute distance $d(X_i, x)$
end
Compute set I containing indices for the k smallest distances $d(X_i, x)$
Return majority label for $\{Y_i \text{ where } i \in I\}$

3.2. Support Vector Machine

Support Vector Machine (SVM) is a linear or non-linear classifier, which is a mathematical function used to distinguish between two different types of objects [35,36]. The SVM can be used for both regression and classification tasks. The basic concept of SVM is that the maximum margin hyperplanes separate the training data to the greatest extent. Figure 2 illustrates a schematic view of SVM.

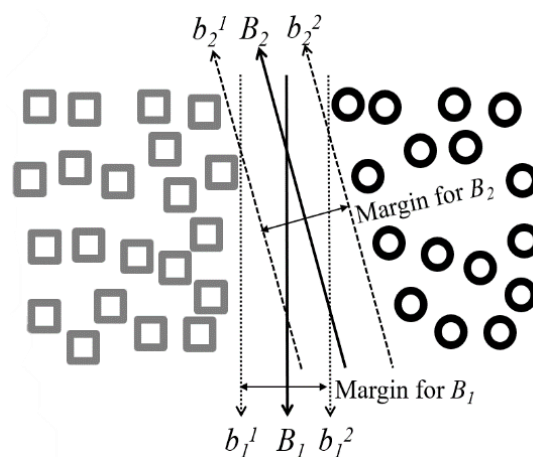


Figure 2. Schematic Concept of SVM.

3.3. Artificial Neural Networks

Standard Artificial Neural Networks (ANNs) are powerful computational models in establishing the relation between variables involving unknown data, particularly for complex non-linear relationships [37,38]. Generally, the ANNs is composed of an input layer which trains the data, hidden layer(s) for computing the weight of the input, and an output layer, where the results of the ANNs are produced. Each layer consists of basic elements called neurons and the neuron is a non-linear algebraic function [39]; since the neurons affect the model performance, determining the number of neurons is important [40,41].

3.4. Standard Extreme Learning Machine

A neural network is a set of connected input/output units where each connection has a weight associated with it. The network learns by adjusting the weights and biases iteratively to minimize the approximate mean square error. In supervised learning, a gradient descent-based learning algorithm is a typical back-propagation (BP) learning algorithm, which updates the network weights and biases in the direction (i.e., gradient direction) in which the error function decreases most rapidly [42]. However,

the learning speed of gradient-based algorithms is generally slow, and the selection of a learning rate is also tedious. These algorithms will be unstable if too large a value is chosen and will converge too slowly if the value is too small. Hence, the Extreme Learning Machine (ELM) algorithm was proposed to overcome these issues.

The ELM [43,44] was originally inspired by biological learning and proposed a modification of the single-layer feedforward network (SLFN), wherein weights and biases are chosen randomly to overcome the challenging issues faced by BP learning algorithms. Unlike other so-called randomness (semi-randomness)-based learning methods/networks [45], all the hidden nodes in the ELM are not only independent of the training data but are also independent of each other. Miche et al. [18] established the universal approximation capability of ELM and its capability for biological learning. Based on the proven theory, the input weights and hidden layer biases of SLFNs can be randomly assigned if the activation functions in the hidden layer are infinitely differentiable, and the standard ELM process is given as follows [46]:

Given a training set $\mathbf{S} = \{(x_i, t_i) | x_i \in R^n, t_i \in R^m, i = 1, \dots, N\}$, activation function $g(x)$, and the number of hidden nodes N .

Step 1: Randomly assign input weight (w_i) and bias ($b_i, i = 1, \dots, N$).

Step 2: Calculate the hidden layer output matrix (H).

Step 3: Calculate the output weight (β),

Where $\beta = H^+ T$, H^+ is the Moore–Penrose generalized inverse of matrix H , and $T = [t_1, \dots, t_N]^T$.

3.5. Online Sequential ELM

To process the standard ELM approach, the parameters of ELM and the entirety of the training data are required for the training step. Therefore, the standard ELM assumes that all the training data are ready prior to the training process. However, in cases of real-time system operation or maintenance tasks, the training data are constantly increased and accumulated by the passage of time or the occurrence of new events.

For this reason, Liang et al. [47] developed the online sequential ELM (OS-ELM), which can train the sequentially accumulated data one-by-one. This learning approach is an appropriate technique for practical applications in which the number of training data or the property of timeliness is not fixed (i.e., training data have a validity period [48]). For instance, in the short-term prediction of stock prices, the training data that are older and less effective are given lower weighting than the recent data. Table 3 presents the pseudo code of OS-ELM [47].

Table 3. Pseudo code of OS-ELM [47].

Input: A training set $\{X, T\} = \{x_i, t_i\}_{i=1}^N$
Output: A trained ELM model
(Standard ELM phase)
Let $k = 0$, Calculate the hidden layer output matrix H_0 using initial training data
Estimate the initial output weight $\beta_0 \rightarrow P_0 = (H_0^T H_0)^{-1}$
(OS-ELM phase)
- When the $(k + 1)$ -th chunk of new data $\{X_{k+1}, T_{k+1}\}$ arrives, update the hidden layer output matrix as $H_{k+1} = [H_k^T, \Delta H_{k+1}^T]^T$
where ΔH_{k+1} is the hidden layer output matrix corresponding to the newly arrived data
- Update the output weights as $\beta_{k+1} = \beta_k + P_{k+1} H_{k+1}^T (T_{k+1} - H_{k+1} \beta_k)$
where $P_{k+1} = P_k - P_k H_{k+1}^T (I + H_{k+1} P H_{k+1}^T)^{-1} H_{k+1} P_k$
- Set $k = k + 1$

3.6. Bidirectional ELM

In the case of the standard ELM, which has a fixed network architecture, the parameter affecting its performance is the number of hidden nodes. However, determining the optimal number of hidden

nodes depends on the sensitivity analysis based on a trial and error approach. To solve this problem, incremental ELM (I-ELM) was proposed by Huang et al. [46]. The I-ELM is an improved version of the standard ELM obtained by adding hidden nodes one-by-one until the expected training accuracy is achieved.

Although I-ELM has been used to determine the appropriate number of hidden nodes, the problem of computation time still remains because I-ELM calculates n output weights one by one when n hidden nodes are used. For these reasons, Yang et al. [49] developed the bidirectional ELM (B-ELM), improving upon the standard ELM algorithm by finding an appropriate number of hidden nodes and enhancing the computing speed and accuracy. The basic concept of B-ELM is the optimization of two main parameters (a_i, b_i) related to the hidden nodes. a_i is the weight vector connecting the input layer to the i th hidden node and b_i is the bias of the i th hidden node. The optimization of these two parameters results in the rapid decrease in residual error. The B-ELM is discussed in detail in the pseudo code given in Table 4.

Table 4. Pseudo code of B-ELM [49].

Input: A training set $\{X, T\} = \{x_i, t_i\}_{i=1}^N \subset R^n \times R$
 The hidden node output function $H(a, b, x)$, the continuous target function f and the hidden node number L , and the expected learning accuracy η .

(Initialization)
 Let the number of hidden nodes $L = 0$, residual error $E = T$, where $t = [t_1, \dots, t_N]$.

(Learning step)
while $L < L_{max}, \|E\| > \eta$
 Increase by one the number of hidden nodes $L: L = L + 1$
 if $L \in \{2n + 1, n \in \mathbb{Z}\}$ **then**
 -Assign random input weight a_L and bias b_L for new hidden node L
 -Calculate the output weight β_L for the new hidden node $\beta_L: \beta_{2n+1} = (e_{2n}, H_{2n+1}^e) / \|H_{2n+1}^e\|^2$
 -Calculate E after adding the new hidden node $L: E = E - H_L \cdot \beta_L$
 end if
 if $L \in \{2n, n \in \mathbb{Z}\}$ **then**
 -Calculate the error feedback function sequence $H_L: H_{2n}^e = e_{2n-1} (\beta_{2n-1})^{-1}$, where, $e_n = f - f_n$.
 -Calculate the input weight a_L , bias b_L and update H_L for the new hidden node L
 -Calculate the output weight β_L for the new hidden node $\beta_L: \beta_{2n} = (e_{2n-1}, H_{2n}^e) / \|H_{2n}^e\|^2$
 -Calculate E after adding the new hidden node $L: E = E - H_L \cdot \beta_L$
 end if
end while

3.7. Weighted ELM

The ELM is a competitive machine learning technique, which is simple in theory and fast in implementation. The network types are “generalized” single hidden layer feedforward networks, which are quite diversified in terms of the variety of feature mapping functions or kernels. To handle data with imbalanced class distribution, a weighted ELM (W-ELM) that can balance the data was proposed [50]. The proposed W-ELM method maintains the advantages of the standard ELM, such as (i) simplicity and convenience, (ii) wide variety of feature mapping functions or kernels, and (iii) capacity for multiclass classification tasks. Moreover, after applying the weighting scheme,

- (1) the W-ELM can deal with data with imbalanced class distribution while maintaining good performance with well-balanced data.
- (2) By assigning different weights for each example according to users’ needs, the W-ELM can be generalized to cost-sensitive learning.

4. Applications, Results, and Discussion

The objective of this study is the development of a novel detection model for the effective detection of cyber-attacks on WDSs by applying various machine learning approaches and determining the

machine learning approach with the best detection performance. To evaluate the performance of each approach in the detection of cyber-attacks on WDSs, the commonly used approaches (i.e., KNN, SVM, ANNs, and ELM) are applied to the BATADAL dataset, simulating various cyber-attack scenarios.

Moreover, to compare the quantitative performance of each algorithm, various performance indices are used. Such performance indices are introduced in the next subsection. The learning approaches are first compared in terms of performance, and the algorithm with the best detection performance is determined. Then, the improved versions of the said learning algorithm are tested and analyzed. All computations were performed using MS Excel 2019, and such machine learning techniques as KNN, ANNs, SVM, ELM and Data Manager were developed using MATLAB 2019 (MathWorks, Inc., Natwick, MA, USA).

4.1. Performance Indices

The performance indices quantitatively measure the prediction accuracy and select the most appropriate model after adjusting the model parameters or model formulation to reduce errors in prediction. In comparing the prediction performance for the various parameters, the model with the least errors was considered as the most accurate model. Such errors were determined by comparing the observed and predicted data.

However, because various performance measures have different prediction errors (e.g., overall error, special point error, and percentage error), the appropriate performance indices must be selected depending on the characteristics of the applied dataset (binary data or real number data). The dataset used in this study is expressed as a binary code (i.e., normal and abnormal condition). Detection performance should be evaluated based on the ability to detect all attacks without raising false alarms.

Therefore, this study has adopted the classification performance measure approach [51] to correctly classify the state of the WDSs under cyber-attack. The classification performance measure approach makes use of a metric referred to as the sensitivity or true positive rate (*TPR*), which is defined as the ratio of the number of time steps correctly classified as “under-attack” to the total number of time steps during which the system is under-attack.

The metric is composed of four variables, namely, the number of true positives (*TP*), number of false positives (*FP*), number of true negatives (*TN*) and number of false negatives (*FN*). These values are then used to calculate: the precision, also known as the positive predictor value (*PPV*); specificity, also known as the true negative rate (*TNR*); and recall, also known as *TPR*. The equations for these variables are given as follows:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

$$\text{Positive predictor value (PPV)} = \frac{TP}{TP + FP} \quad (2)$$

$$\text{True Positive rate (TPR)} = \frac{TP}{TP + FN} \quad (3)$$

$$\text{True Negative rate (TNR)} = \frac{TN}{TN + FP} \quad (4)$$

To facilitate the comparison among the reported algorithms, this study combined *PPV* and *TPR* into the F1 score, which gives the same importance to both metrics. The F1 score is defined as the harmonic average of the two metrics, and is expressed in the following equation:

$$F1 = 2 \times \frac{PPV \times TPR}{PPV + TPR} \quad (5)$$

Moreover, the time-to-detection (TTD) value is the difference between the time when an attack starts and when it is first flagged as an under-attack event. It is used to calculate the score of time-to-detection (S_{TTD}) for effective comparison of all detection models under different attack scenarios. These variables are defined in the following equations:

$$\text{Time to detection (TTD)} = t_d - t_o (0 \leq TTD \leq \Delta t) \quad (6)$$

$$\text{Score of time to detection (S}_{TTD}) = 1 - \frac{1}{n_a} \sum_{i=1}^{n_a} \frac{TTD_i}{\Delta t_i} \quad (7)$$

where Δt is the total duration of the attack, n_a is the number of attacks contained in a dataset, TTD_i is the TTD relative to the i th attack, and Δt_i is the corresponding duration. The overall performance score S is used to rank the algorithms given as follows:

$$S = \gamma \times S_{TTD} + (1 - \gamma) \times S_{CLF} \quad (8)$$

where S_{CLF} is the score of classification performance and is calculated by $S_{CLF} = (TPR + TNR)/2$, and the weight factor for detection importance of time detection or classification performance is γ (set to 0.5 in this study). All metrics described in this section vary between 0 and 1, with 0 representing the poorest performance and 1 representing perfect accuracy.

4.2. Comparison of Prediction Results for the Various Standard Classification Approaches

To determine the algorithm that exhibits the best detection performance, a comparison of the prediction results was conducted for the various original classification approaches dealing with WDS cyber-attack situations. Before the performance comparison, a sensitivity analysis of the parameters of each algorithm (e.g., training function, adaptation learning function, performance function, number of layers, number of neurons, and transfer function) was first performed by comparing the model performances depending on the parameter variation (see Table 5).

Table 5. Selected parameters result for the parameter sensitivity analysis of applied machine learning algorithms.

Algorithms	User Parameters
KNN	-Number of nearest neighbors = 5
ANNs	-Training function = Bayesian regularization back-propagation (TRAINBR) -Adaptation learning function = Bias learning function (LEARNGD) -Transfer function = Log-sigmoid (LOGSIG) -Number of layers = 3 -Number of neurons = 10 -70% training, 15% validation, and 15% test datasets, respectively
SVM	-Mathematical functions = Kernel Functions -Polynomial order = 3
ELM	-Activation function = Sine function -Number of hidden neurons = 52

To show the outperformed algorithm, this study performed a sensitivity analysis of the parameters of each algorithm. Based on the sensitivity analysis process, we found out appropriate parameters on each algorithm. Therefore, the performance comparison performed fairly. In the sensitivity analysis process, the parameters were changed within the range. The ANNs considered six parameters under each parameter's range (training functions: among 19; adaptation learning functions: among 5; transfer functions: among 3; number of layers: 1~10 (gap 1); number of neurons: 1~50 (gap 5); training probability: 40~80% (gap 10%)).

The SVM considered two parameters under each parameter's range (mathematical functions: among 4; polynomial order: 1~10% (gap 0.5)), while the ELM considered two parameters under each parameter's range (activation functions: among 5; number of hidden neurons: 10~150 (gap 1)). The Table 5 shows the results for the sensitivity analysis of the parameters of each algorithm.

Table 6 represents the performance comparison results for the original versions of the learning algorithms. A visual detection comparison of the normal status and the under-attack conditions is illustrated in the bar charts given in Figure 3. The applied cyber-attack data comprise training/testing input and output data. The output data are configured as binary codes (i.e., normal condition and under-attack condition) based on the various hydraulic results of the system components (i.e., water level of tanks, status and water flow of pumps, status and water flow of valves, and pressure of junctions). It can be observed from Table 6 that an accuracy value according to the four variables (i.e., *TP*, *FP*, *TN*, and *FN*) can be used to evaluate the algorithms' total detection ability regardless of whether the condition is set as normal or under-attack.

Table 6. Performance comparison results for original version of machine learning algorithms.

Performance Indices	KNN	ANNs	SVM	ELM
S	0.418	0.749	0.754	0.891
SCLF	0.512	0.708	0.786	0.841
S _{TTD}	0.323	0.789	0.722	0.941
F1	0.048	0.558	0.694	0.764
TNR	1.000	0.952	0.967	0.959
TPR	0.025	0.464	0.604	0.722
PPV	1.000	0.700	0.815	0.810
Accuracy	0.810	0.857	0.876	0.913
TP	10	189	246	294
FP	0	81	56	69
TN	1682	1601	1626	1613
FN	397	218	161	113

The detection accuracy values of all the applied algorithms were over 80%, which is quite high (see Table 6). However, the KNN exhibited a significantly lower *TPR* than the other algorithms, only detecting cyber-attacks 10 times (*TP*). For the ANNs, most of the performance indices, except *S_{TTD}*, were of similar values or less than that of the SVM. Moreover, among the total events (normal conditions: 1682; under-attack conditions: 407), ANNs exhibited higher values of *FP* and *FN* than the SVM, amounting to 81 and 218, respectively. The ANNs exhibited a high false detection rate. However, the value of *S_{TTD}* seen in the ANNs was 0.789, which was higher than that in SVM.

The *S_{TTD}* is defined as the detection ability at the beginning of the “under-attack” conditions. Although the ANNs performed better in the detection of the initial cyber-attack event in comparison with the SVM, the overall detection ability of ANNs was worse. This shows that it lacks the ability to detect cyber-attacks consistently after the initial cyber-attack. Figure 3b,c illustrate the difference in the detection ability of the ANNs and SVM through visual detection comparison. In Figure 3, the red bars represent the prediction data for under-attack conditions and the black bars indicate the observed data. In the detection history of ANNs (Figure 3b), most of the red bars are located in the initial event corresponding to each attack scenario except two cases (i.e., 2nd and 5th attack scenarios). However, the SVM detected four initial events among seven scenarios, and subsequently, most of the under-attack events were predicted constantly.

Based on the comparison among the machine learning algorithms, the performance of ELM is outstanding in all aspects, particularly in detection accuracy and initial attack event detection. As depicted in Figure 3d, the ELM detected all the cyber-attack scenarios and even detected most of the initial attack events in the seven scenarios. However, some of the attack events were not detected and there were some false detections around the attack events. This proves that although the ELM was able to consistently and accurately detect most of the attacks, it failed to do so consistently.

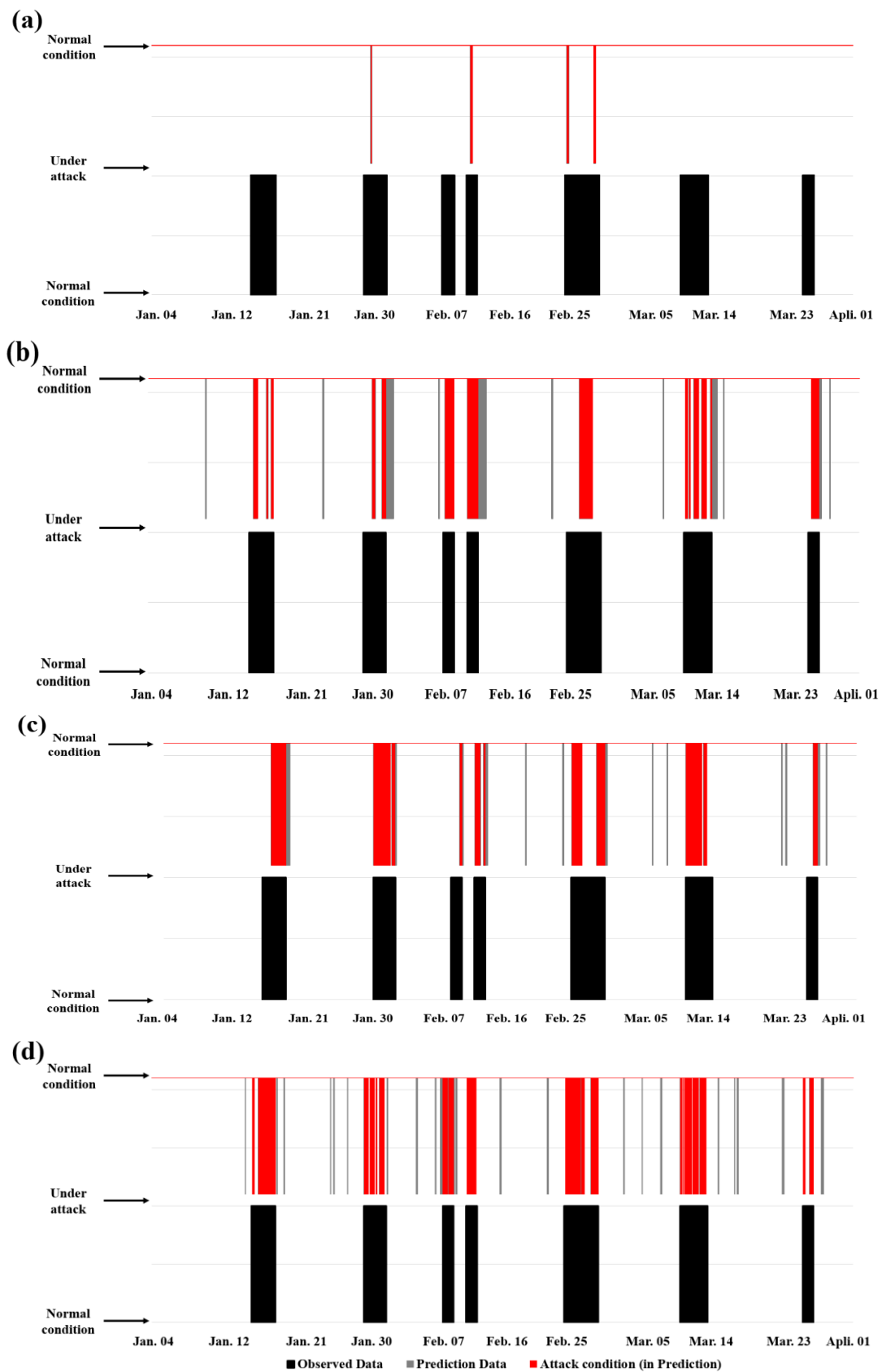


Figure 3. Performance comparison between observed and prediction data for each detection algorithm: (a) KNN, (b) ANNs, (c) SVM, and (d) ELM.

4.3. Comparison of Prediction Results of Improved Versions of ELM

As represented in Section 4.2, the ELM was identified to be the best machine learning approach. This section compares the improved versions of the ELM algorithm, namely, the online sequential ELM, bidirectional ELM, and weighted ELM, with respect to the prediction quality and performance measures. The results are analyzed according to their algorithm's characteristics. Because the performance of ELM affects the values of its parameters (i.e., activation function and number of hidden neurons), to evaluate the performance, the validation process of such parameters is necessary [52].

In response to this, the sensitivity analysis of the parameters was conducted for the improved version of ELM algorithms. Based on the parameter sensitivity analysis, the applied activation function and number of hidden neurons of each algorithm are as follows: OS-ELM: triangular basis (Tribas) function and 10; B-ELM: sine function, and 10; W-ELM: sigmoid function and 75. The cyber-attack prediction results are presented in Table 7 and Figure 4.

Table 7. Comparison of performance results among the improved versions of ELM.

Performance Indices	ELM	OS-ELM	B-ELM	W-ELM
S	0.891	0.910	0.900	0.906
S_{CLF}	0.841	0.880	0.865	0.832
S_{TTD}	0.941	0.940	0.935	0.980
F1	0.764	0.806	0.768	0.744
TNR	0.959	0.952	0.958	0.951
TPR	0.722	0.808	0.732	0.713
PPV	0.810	0.804	0.808	0.777
Accuracy	0.913	0.924	0.914	0.904
TP	294	329	298	290
FP	69	80	71	83
TN	1613	1602	1611	1599
FN	113	78	109	117

Table 7 highlights the three improved versions of ELM and the standard ELM, wherein all variants, including the standard ELM, achieved a ranking score S higher than (or close to) 0.9. The results of OS-ELM show the best overall performance ($S = 0.910$). The OS-ELM has the top score with respect to the classification score S_{CLF} . Moreover, the detection history depicted in Figure 4a shows that all attacks are immediately detected, with the exception of the first attack scenario, which is detected a few hours after its starting time. However, in comparison with the other improved algorithms, false alarms are more likely to occur before an attack occurs. This is because the OS-ELM was developed considering real-time system operation or maintenance tasks, wherein the training data are accumulated over time. Therefore, it can be shown that the time difference results in false prediction as most of the false detections are generated intensively before an attack occurs. The performance of B-ELM is close to that of the standard ELM and W-ELM with respect to the S metric and identifying TTD. However, it is more likely to generate false alarms. Unlike other variants of ELM, B-ELM increases the user convenience in terms of determining the appropriate number of hidden nodes while providing accurate prediction results without the need to conduct sensitivity analysis for the number of hidden neurons.

The W-ELM has a similar ranking score S to that of OS-ELM and B-ELM. The S_{CLF} value of W-ELM is lower than that of the others, resulting in a lower TPR and higher FN . All of these lead to a score of S equal to 0.906. Moreover, the W-ELM algorithm is shown to have the least timing error, as presented in Table 7. This implies that although almost all the starting attack points can be detected using the W-ELM, it is also prone to FPs. This may be due to the weight of data in the training process. The weight of abnormal cases is affected by the cyber-attack detection. For this reason, most of the false predictions for the W-ELM occur close to the attack conditions.

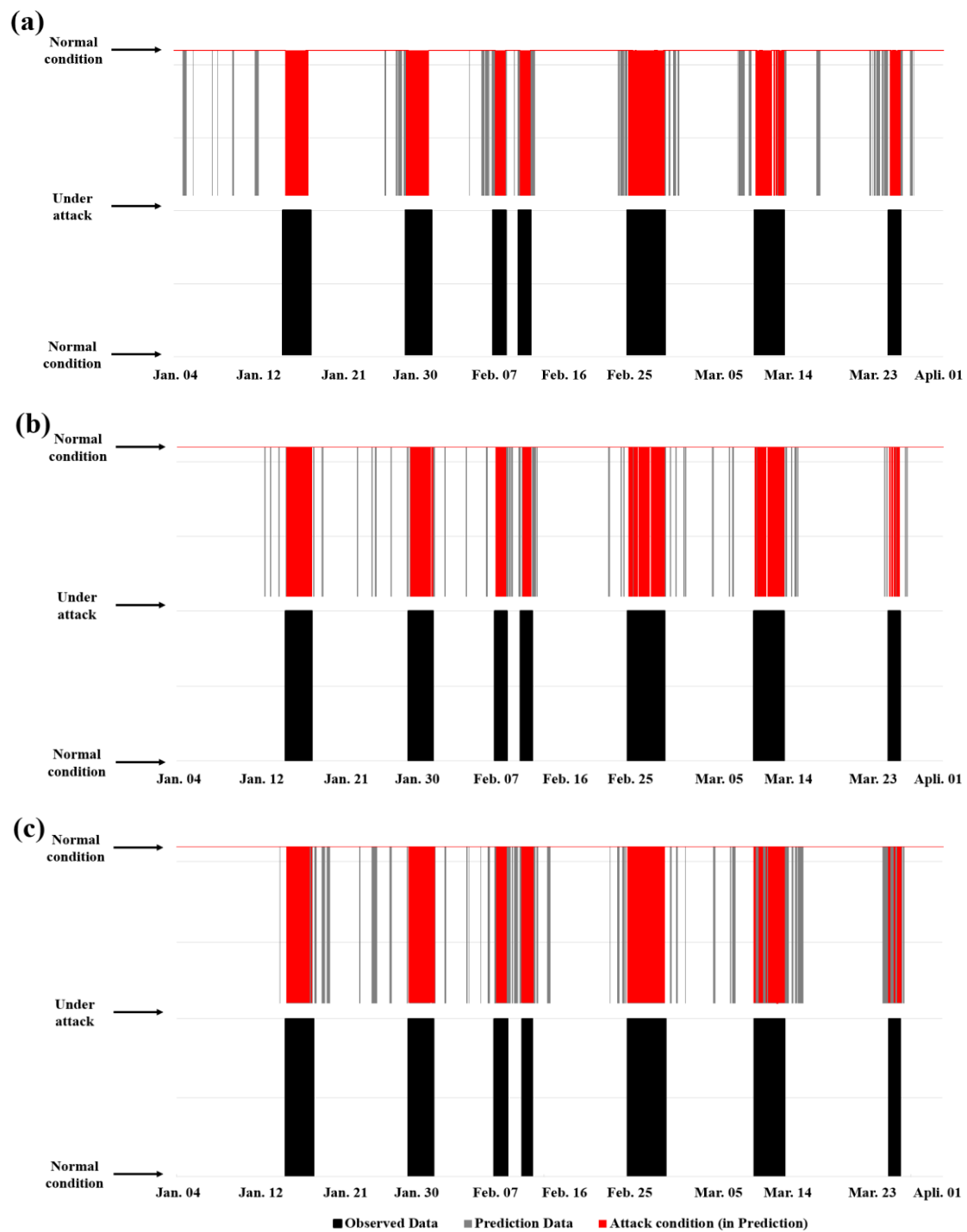


Figure 4. Performance comparison between observed and prediction data for the improved versions of ELM: (a) OS-ELM, (b) B-ELM, (c) W-ELM.

5. Conclusions and Future Studies

This study proposes a novel detection model, which uses machine learning approaches, for the detection of cyber-attacks on WDSs. To propose a novel cyber-attack detection model in WDSs, this study performed two types of analyses. First, the machine learning approach with the most suitable detection performance was identified and analyzed. For this analysis, the most commonly used and well-known machine learning algorithms (i.e., KNN, SVM, ANNs, and ELM) were compared to determine the one with the best detection performance. Second, by considering the improved

version of the outstanding algorithm among the compared algorithms, the characteristics of the improved algorithms, in terms of the cyber-attack detection problem, were analyzed. Moreover, various performance indices were used to compare the quantitative performance of each algorithm and correctly classify the state of the WDSs under cyber-attack situations.

According to the performance analysis of the improved versions of ELM in this study, the three ELM algorithms have an outstanding performance compared with the other machine learning approaches (e.g., KNN, ANNs, and SVM) in the aspect of the score S , which reflects overall performance. However, through the applied problem in this study and the characteristics of each ELM algorithm, this study can derive several limitations. The initial version of ELM was developed to increase process speed, accuracy and to solve large real-world problems. Therefore, according to the simulation results of this study, the performance of ELMs was outperformed, however, compared to the other algorithms applied in this study, the ratio of the false positive detection was relatively high.

The trend of false positive detection differs depending on the algorithm's characteristics, such as when the false predictions occur before the attack (OS-ELM), or when most of the false predictions occur near the attack conditions (W-ELM). One practical solution for solving this problem may be using some heuristic pre-knowledge information from the urban water systems and hydraulic system to make the classifier more intelligent. This helps the classifier to use some of its pre-knowledge to accurately predict the right condition. In addition, utilizing fuzzy logic (fuzzy theory) along with the ELM makes decision making much easier.

Similarly, the various ELMs proposed have several limitations in terms of algorithm improvement (e.g., reducing false detection and improving true detection probability) and application aspects (e.g., image training problems, abnormal condition detection problems, time series problems, and regression problems). Although this study has the above limitations, it analyzed the characteristics of various types of ELM algorithms and the application problems of the various ELMs; this research serves as a good case study of ELMs to extend their applications to different and various fields.

Moreover, the generated prediction results and analyses are expected to benefit the development of new ELM approaches, considering the thorough evaluation of each algorithm. Future studies may refer to the conclusions of this study to develop highly accurate algorithms. In doing so, one must not only consider benchmark problems but various other types of problems as well, such as (i) training data configuration: binary, real number, integer, and mix-integer; (ii) data collection type: sequencing data, prepared whole data; and (iii) data characteristics: continuous data (i.e., time series) versus discrete data. Besides, other improved/hybrid versions of ELM, particularly convolutional neural networks used for feature learning along with ELM as a classifier, may enhance the accuracy of the obtained results.

Author Contributions: Conceptualization, A.S. and J.H.K.; Data curation, Y.H.C.; Methodology, Y.H.C.; Supervision, J.H.K.; Writing—original draft, Y.H.C.; Writing—review and editing, A.S. and J.H.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by The National Research Foundation of Korea, grant number 2019R1A2B5B03069810.

Acknowledgments: This work was supported by a grant from The National Research Foundation of Korea, funded by the Korean government (MSIP).

Conflicts of Interest: There is no conflict of interest and the source of funding, including the grant number for this paper, was declared.

References

1. Spellman, F.R. *Handbook of Water and Wastewater Treatment Plant Operations*; CRC Press: Boca Raton, FL, USA, 2013.
2. Bobat, A.; Gezgin, T.; Aslan, H. The SCADA system applications in management of Yuvacik Dam and Reservoir. *Desalin. Water Treat.* **2014**, *54*, 2108–2119. [[CrossRef](#)]
3. Rasekh, A.; Hassanzadeh, A.; Mulchandani, S.; Modi, S.; Banks, M.K. Smart Water Networks and Cyber Security. *J. Water Resour. Plan. Manag.* **2016**, *142*, 01816004. [[CrossRef](#)]

4. Moyer, J.; Dakin, R.; Hewman, R.; Groves, D. The Case for Cyber Security in the Water Sector. *J. Am. Water Work. Assoc.* **2009**, *101*, 30–32. [[CrossRef](#)]
5. ICS-CERT. NCCIC/ICS-CERT Year in Review: FY 2015; Report No. 15e50569; U.S. Department of Homeland Security Industrial Control Systems-Cyber Emergency Response Team: Washington, DC, USA, 2016.
6. Slay, J.; Miller, M. Lessons learned from the maroochy water breach. In Proceedings of the International Conference on Critical Infrastructure Protection, Hanover, NH, USA, 19–21 March 2007; pp. 73–82.
7. Ostfeld, A.; Salomons, E.; Ormsbee, L.; Uber, J.G.; Bros, C.M.; Kalungi, P.; Burd, R.; Zazula-Coetzee, B.; Belrain, T.; Kang, D.; et al. Battle of the Water Calibration Networks. *J. Water Resour. Plan. Manag.* **2012**, *138*, 523–532. [[CrossRef](#)]
8. Wu, Y.; Liu, S.; Wu, X.; Liu, Y.; Guan, Y. Burst detection in district metering areas using a data driven clustering algorithm. *Water Res.* **2016**, *100*, 28–37. [[CrossRef](#)] [[PubMed](#)]
9. Gong, W.; Suresh, M.A.; Smith, L.; Ostfeld, A.; Stoleru, R.; Rasekh, A.; Banks, M.K. Mobile sensor networks for optimal leak and backflow detection and localization in municipal water networks. *Environ. Model. Softw.* **2016**, *80*, 306–321. [[CrossRef](#)]
10. Choi, Y.H.; Kim, J.H. Topological and mechanical redundancy-based optimal design of water distribution systems in many-objective optimization. *Eng. Optim.* **2019**, 1–18. [[CrossRef](#)]
11. Jung, D.; Lansey, K.E.; Choi, Y.H.; Kim, J.H. Robustness-based optimal pump design and scheduling for water distribution systems. *J. Hydroinformatics* **2015**, *18*, 500–513. [[CrossRef](#)]
12. Choi, Y.H.; Jung, D.; Jun, H.; Kim, J.H. Improving Water Distribution Systems Robustness through Optimal Valve Installation. *Water* **2018**, *10*, 1223. [[CrossRef](#)]
13. Yang, D.; Usynin, A.; Hines, J.W. Anomaly-based intrusion detection for SCADA systems. In Proceedings of the 5th International Topical Meeting on Nuclear Plant Instrumentation, Control and Human Machine Interface Technologies (npic&hmit 05), Albuquerque, NM, USA, 12–16 November 2006; pp. 12–16.
14. Cleveland, F.M. Cyber security issues for advanced metering infrastructure (AMI). In Proceedings of the 2008 IEEE Power and Energy Society General Meeting—Conversion and Delivery of Electrical Energy in the 21st Century, Pittsburgh, PA, USA, 20–24 July 2008; pp. 1–5. [[CrossRef](#)]
15. Xie, L.; Mo, Y.; Sinopoli, B. False Data Injection Attacks in Electricity Markets. In Proceedings of the 2010 First IEEE International Conference on Smart Grid Communications, Gaithersburg, MD, USA, 4–6 October 2010; pp. 226–231.
16. Kim, T.T.; Poor, H.V. Strategic Protection Against Data Injection Attacks on Power Grids. *IEEE Trans. Smart Grid* **2011**, *2*, 326–333. [[CrossRef](#)]
17. Liu, X.; Liu, X.; Li, Z. Cyber Risk Assessment of Transmission Lines in Smart Grids. *Energies* **2015**, *8*, 13796–13810. [[CrossRef](#)]
18. Miche, Y.; Sorjamaa, A.; Bas, P.; Simula, O.; Jutten, C.; Lendasse, A. OP-ELM: Optimally Pruned Extreme Learning Machine. *IEEE Trans. Neural Netw.* **2009**, *21*, 158–162. [[CrossRef](#)]
19. Negrete-Pincetic, M.; Yoshida, F.; Gross, G. Towards quantifying the impacts of cyber attacks in the competitive electricity market environment. *IEEE Buchar. Power Tech.* **2009**, 1–8. [[CrossRef](#)]
20. Kosut, O.; Jia, L.; Thomas, R.J.; Tong, L. Limiting false data attacks on power system state estimation. In Proceedings of the 2010 44th Annual Conference on Information Sciences and Systems (CISS), Princeton, NJ, USA, 17–19 March 2010; pp. 1–6.
21. Esmalifalak, M.; Shi, G.; Han, Z.; Song, L. Bad Data Injection Attack and Defense in Electricity Market Using Game Theory Study. *IEEE Trans. Smart Grid* **2013**, *4*, 160–169. [[CrossRef](#)]
22. Goh, J.; Adepu, S.; Tan, M.; Lee, Z.S. Anomaly Detection in Cyber Physical Systems Using Recurrent Neural Networks. In Proceedings of the 2017 IEEE 18th International Symposium on High Assurance Systems Engineering (HASE), Singapore, 12–14 January 2017; pp. 140–145.
23. Inoue, J.; Yamagata, Y.; Chen, Y.; Poskitt, C.M.; Sun, J. Anomaly Detection for a Water Treatment System Using Unsupervised Machine Learning. In Proceedings of the 2017 IEEE International Conference on Data Mining Workshops (ICDMW), New Orleans, LA, USA, 18–24 November 2017; pp. 1058–1065.
24. Kravchik, M.; Shabtai, A. Detecting Cyber Attacks in Industrial Control Systems Using Convolutional Neural Networks. In Proceedings of the 2018 Workshop on Cyber-Physical Systems Security and Privacy-CPS-SPC '18, Toronto, ON, Canada, 19 October 2018; pp. 72–83.

25. Amin, S.; Litrico, X.; Sastry, S.; Bayen, A.M. Cyber Security of Water SCADA Systems—Part I: Analysis and Experimentation of Stealthy Deception Attacks. *IEEE Trans. Control. Syst. Technol.* **2013**, *21*, 1963–1970. [\[CrossRef\]](#)
26. Perelman, L.; Amin, S. A network interdiction model for analyzing the vulnerability of water distribution systems. In Proceedings of the 3rd International Conference on Pervasive Technologies Related to Assistive Environments-PETRA '10, Samos, Greece, 23–25 June 2014; pp. 135–144.
27. Almalawi, A.; Fahad, A.; Tari, Z.; Alamri, A.; Alghamdi, R.; Zomaya, A.Y. An Efficient Data-Driven Clustering Technique to Detect Attacks in SCADA Systems. *IEEE Trans. Inf. Forensics Secur.* **2016**, *11*, 893–906. [\[CrossRef\]](#)
28. Taormina, R.; Galelli, S.; Tippenhauer, N.O.; Salomons, E.; Ostfeld, A. Characterizing Cyber-Physical Attacks on Water Distribution Systems. *J. Water Resour. Plan. Manag.* **2017**, *143*, 04017009. [\[CrossRef\]](#)
29. Rossman, L.A. *EPANET 2: Users' Manual*; BiblioGov, U.S. Environmental Protection Agency: Washington, DC, USA, 2000.
30. Taormina, R.; Galelli, S. Deep-Learning Approach to the Detection and Localization of Cyber-Physical Attacks on Water Distribution Systems. *J. Water Resour. Plan. Manag.* **2018**, *144*, 04018065. [\[CrossRef\]](#)
31. Taormina, R.; Galelli, S.; Tippenhauer, N.O.; Salomons, E.; Ostfeld, A.; Eliades, D.G.; Aghashahi, M.; Sundararajan, R.; Pourahmadi, M.; Banks, M.K.; et al. Battle of the Attack Detection Algorithms: Disclosing Cyber Attacks on Water Distribution Networks. *J. Water Resour. Plan. Manag.* **2018**, *144*, 04018048. [\[CrossRef\]](#)
32. Altman, N.S. An Introduction to Kernel and Nearest-Neighbor Nonparametric Regression. *Am. Stat.* **1992**, *46*, 175. [\[CrossRef\]](#)
33. Bhatia, N. Survey of nearest neighbor techniques. *arXiv* **2010**, arXiv:1007.0085.
34. Tay, B.; Hyun, J.K.; Oh, S. A Machine Learning Approach for Specification of Spinal Cord Injuries Using Fractional Anisotropy Values Obtained from Diffusion Tensor Images. *Comput. Math. Methods Med.* **2014**, *2014*, 1–8. [\[CrossRef\]](#)
35. Vapnik, V. *Statistical Learning Theory*; Wiley: New York, NY, USA, 1998.
36. Cho, D. Mixed-effects LS-SVM for longitudinal data. *J. Korean Data Inf. Sci. Soc.* **2010**, *21*, 363–369.
37. Gallant, S.I.; Gallant, S.I. *Neural Network Learning and Expert Systems*; A Bradford Book; MIT Press: Cambridge, MA, USA, 1993.
38. Smith, M. *Neural Networks for Statistical Modelling*; John Wiley & Sons: New York, NY, USA, 1994; p. 235.
39. Dreyfus, G.; Martinez, J.-M.; Samuelides, M.; Gordon, M.B.; Badran, F.; Thiria, S.; Herault, L. *Reseaux de Neurones: Methodologie et Applications*; Editions Eyrolles: Paris, France, 2002.
40. Karunanithi, N.; Grenney, W.J.; Whitley, D.; Bovee, K. Neural Networks for River Flow Prediction. *J. Comput. Civ. Eng.* **1994**, *8*, 201–220. [\[CrossRef\]](#)
41. ASCE Task Committee on Application of Artificial Neural Networks in Hydrology. Artificial Neural Networks in Hydrology. II: Hydrologic Applications. *J. Hydrol. Eng.* **2000**, *5*, 124–137. [\[CrossRef\]](#)
42. Hagan, M.T.; Demuth, H.B.; Beale, M. *Neural Network Design*; Thomson Learning, Oklahoma State University: Stillwater, OK, USA, 2002.
43. Huang, G.B.; Zhu, Q.Y.; Siew, C.K. Extreme learning machine: A new learning scheme of feedforward neural networks. In Proceedings of the 2004 IEEE International Joint Conference on Neural Networks (IEEE Cat. No. 04CH37541), Budapest, Hungary, 25–29 July 2004; Volume 2, pp. 985–990. [\[CrossRef\]](#)
44. Li, M.-B.; Huang, G.-B.; Saratchandran, P.; Sundararajan, N. Fully complex extreme learning machine. *Neurocomputing* **2005**, *68*, 306–314. [\[CrossRef\]](#)
45. Igel'nik, B.; Pao, Y.-H. Stochastic choice of basis functions in adaptive function approximation and the functional-link net. *IEEE Trans. Neural Netw.* **1995**, *6*, 1320–1329. [\[CrossRef\]](#)
46. Huang, G.-B.; Zhu, Q.-Y.; Siew, C.-K. Extreme learning machine: Theory and applications. *Neurocomputing* **2006**, *70*, 489–501. [\[CrossRef\]](#)
47. Liang, N.-Y.; Huang, G.-B.; Saratchandran, P.; Sundararajan, N. A Fast and Accurate Online Sequential Learning Algorithm for Feedforward Networks. *IEEE Trans. Neural Netw.* **2006**, *17*, 1411–1423. [\[CrossRef\]](#)
48. Zhao, J.; Wang, Z.; Park, D.S. Online sequential extreme learning machine with forgetting mechanism. *Neurocomputing* **2012**, *87*, 79–89. [\[CrossRef\]](#)
49. Yang, Y.; Wang, Y.; Yuan, X. Bidirectional Extreme Learning Machine for Regression Problem and Its Learning Effectiveness. *IEEE Trans. Neural Networks Learn. Syst.* **2012**, *23*, 1498–1505. [\[CrossRef\]](#) [\[PubMed\]](#)
50. Zong, W.; Huang, G.-B.; Chen, Y. Weighted extreme learning machine for imbalance learning. *Neurocomputing* **2013**, *101*, 229–242. [\[CrossRef\]](#)

51. Wang, G.; Hao, J.; Ma, J.; Huang, L. A new approach to intrusion detection using Artificial Neural Networks and fuzzy clustering. *Expert Syst. Appl.* **2010**, *37*, 6225–6232. [[CrossRef](#)]
52. Cao, W.; Gao, J.; Ming, Z.; Cai, S. Some tricks in parameter selection for extreme learning machine. In Proceedings of the IOP Conference Series: Materials Science and Engineering, HI, USA, 30 August–2 September 2017; Volume 261, p. 012002.

Publisher’s Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).