



**Michigan
Technological
University**

Michigan Technological University
Digital Commons @ Michigan Tech

Michigan Tech Publications

1-30-2022

Long Short-Term Memory Based Subsurface Drainage Control for Rainfall-Induced Landslide Prevention

Aynaz Biniyaz

Michigan Technological University, abiniyaz@mtu.edu

Behnam Azmoon

Michigan Technological University, bazmoon@mtu.edu

Ye Sun

University of Virginia

Zhen Liu

Michigan Technological University

Follow this and additional works at: <https://digitalcommons.mtu.edu/michigantech-p>



Part of the [Civil and Environmental Engineering Commons](#)

Recommended Citation

Biniyaz, A., Azmoon, B., Sun, Y., & Liu, Z. (2022). Long Short-Term Memory Based Subsurface Drainage Control for Rainfall-Induced Landslide Prevention. *Geosciences (Switzerland)*, 12(2). <http://doi.org/10.3390/geosciences12020064>

Retrieved from: <https://digitalcommons.mtu.edu/michigantech-p/15723>



Follow this and additional works at: <https://digitalcommons.mtu.edu/michigantech-p>



Part of the [Civil and Environmental Engineering Commons](#)

Article

Long Short-Term Memory Based Subsurface Drainage Control for Rainfall-Induced Landslide Prevention

Aynaz Biniyaz ¹, Behnam Azmoon ¹, Ye Sun ² and Zhen Liu ^{1,*}

¹ Department of Civil, Environmental and Geospatial Engineering, Michigan Technological University, Houghton, MI 49931, USA; abiniyaz@mtu.edu (A.B.); bazmoon@mtu.edu (B.A.)

² Department of Mechanical and Aerospace Engineering, University of Virginia, Charlottesville, VA 22904, USA; dzv7sg@virginia.edu

* Correspondence: zhenl@mtu.edu

Abstract: Subsurface drainage has been widely accepted to mitigate the hazard of landslides in areas prone to flooding. Specifically, the use of drainage wells with pumping systems has been recognized as an effective short-term solution to lower the groundwater table. However, this method has not been well considered for long-term purposes due to potentially high labor costs. This study aims to investigate the idea of an autonomous pumping system for subsurface drainage by leveraging conventional geotechnical engineering solutions and a deep learning technique—Long-Short Term Memory (LSTM)—to establish a geotechnical cyber-physical system for rainfall-induced landslide prevention. For this purpose, a typical soil slope equipped with three pumps was considered in a computer simulation. Forty-eight cases of rainfall events with a wide range of varieties in duration, total rainfall depths, and different rainfall patterns were generated. For each rainfall event, transient seepage analysis was performed using newly proposed Python code to obtain the corresponding pump's flow rate data. A policy of water pumping for maintaining groundwater at a desired level was assigned to the pumps to generate the data. The LSTM takes rainfall event data as the input and predicts the required pump's flow rate. The results from the trained model were validated using evaluation metrics of root mean square error (RMSE), mean absolute error (MAE), and R^2 . The R^2 -scores of 0.958, 0.962, and 0.954 for the predicted flow rates of the three pumps exhibited high accuracy of the predictions using the trained LSTM model. This study is intended to make a pioneering step toward reaching an autonomous pumping system and lowering the operational costs in controlling geosystems.

Keywords: landslide prevention; disaster resilience; geotechnical cyber-physical systems; long short-term memory; deep learning; transient flow



Citation: Biniyaz, A.; Azmoon, B.; Sun, Y.; Liu, Z. Long Short-Term Memory Based Subsurface Drainage Control for Rainfall-Induced Landslide Prevention. *Geosciences* **2022**, *12*, 64. <https://doi.org/10.3390/geosciences12020064>

Academic Editors: Manolis Veveakis, Mohamed Shahin and Jesus Martinez-Frias

Received: 23 November 2021

Accepted: 26 January 2022

Published: 30 January 2022

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



Copyright: © 2022 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 (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Natural disasters cause severe human and economic losses all over the world [1]. Landslides account for approximately 5% of natural disasters and result in economic losses of \$1.6 to \$3.2 billion in the United States and approximately 25–50 deaths annually [2]. Researchers have conducted landslide susceptibility assessments in which landslide susceptibility is defined as the probability of landslide occurrence in an area considering various geoenvironmental conditions, and rainfall has been identified as one of the main factors influencing landslide occurrence [3–7]. The 2010 Maierato landslide [8], the 2014 Oso landslide [9], the 2018 Mengdong landslides [10], and annually occurring landslides in Rize, Turkey [11] are examples of rainfall-induced landslides. The frequency, duration, and intensity of rainfall events have been affected by climate change in recent decades, leading to an increase in landslides [12–15]. Intense and prolonged precipitation can cause a rise in the groundwater table, which decreases the stability of the slope. Additionally, the rain infiltrating the slope increases the degree of saturation and the unit weight of soil. The resultant increase in soil saturation also causes a decrease in the matric suction and shear

strength of the unsaturated soil [16–19], which can adversely impact the safety of geosystems such as slopes. The magnitude and rate of the impact on the soil properties and the safety factor of a slope depend on the intensity and the duration of the rain [20,21]. Slope failures induced by rainfall events and the above mechanisms can cause massive economic losses and fatalities [2]; for example, [22] reported that monsoon flooding and landslides in the 2019 summer impacted more than 7 million people in Nepal, India, and Bangladesh.

To mitigate landslide hazards in areas prone to flooding, drainage has been widely used as a stabilization method [23–26]. Lowering the groundwater table using subsurface drainage systems increases slope stability. The lowered water table reduces the driving forces, which increases the factor of safety (FS). Most drainage systems are designed to use the gravity flow to reduce costs. In some cases, a pumping system might be necessary to remove water from deep drainage wells due to an inadequate gravity force [27].

Deep wells equipped with pumps and well-point systems are two pumping-based techniques that can lower the groundwater level in shallow and deep excavations, respectively [28]. For example, deep wells equipped with pumps were utilized as a primary solution to stabilize a slope near Genoa, Italy, in 1987 [29,30]. Such systems are also applicable in mining projects to reduce the risk of liquefaction of mine tailings due to earthquakes [31]. The operation of a well-point system requires a specialist operator to tune the flow valves to prevent the water level from falling below the top of the screen, as any air entering the system might cause a pump malfunction. Regular checks may also be required in pumping-based techniques to control seepage in sites subject to flooding [28,32]. The high operational cost of such systems is the primary factor preventing pumping systems from developing into long-term solutions for dewatering [28].

To overcome such problems, the goal of this study is to investigate the idea of an autonomous pumping system for subsurface drainage by leveraging conventional geotechnical engineering solutions and deep learning techniques to establish a geotechnical cyber-physical system (CPS) for rainfall-induced landslide prevention. To the best of our knowledge, this study is the first to explore the application of deep learning techniques in practical geotechnical CPSs. Specifically, we propose to use the Long Short-Term Memory (LSTM) recurrent network to learn the control policy of water pumping from the observed data during rainfall events so the trained LSTM can be used to run pumps for desired control outcomes with the predicted pump's flow rate. A virtual environment was created to generate the data for training and testing of the proposed LSTM model, which has been validated with commercial software. Accordingly, Python code was developed to implement a transient seepage model for a typical soil slope equipped with three pumps and subjected to rainfall events. The proposed numerical simulation framework was integrated with deep learning, which cannot be achieved with commercially available software for transient seepage analysis. The LSTM model was then trained to take the rainfall data as input and predict the required pump's flow rate data.

The proposed method can identify unnoticed knowledge in observed data that can enable autonomous water pumping in geosystems toward data-driven resilience. The proposed system can significantly reduce operational costs and improve safety by replacing human-controlled and managed geosystems. The main scientific contributions can be summarized as follows:

- **Geotechnical CPS:** This study is the first to leverage LSTMs for subsurface drainage control for landslide prevention as a concept. The established geotechnical CPS in this study can be used for establishing autonomous pumping systems to increase safety and to reduce the operational costs of geosystems.
- **Data-driven disaster resilience:** The proposed method can learn from data and experiences by integrating deep learning techniques into geotechnical engineering solutions, which is among the first to control an infrastructure system involving complex physics in geomaterials and to provide unique interventions to hazard prevention toward data-driven disaster resilience.

- **Physics-based model for data generation:** We establish a new physics-based model for the transient seepage analysis of geosystems considering precipitation and pumping, which is needed to stimulate the behavior of such CPSs for improved drainage and landslide prevention. The developed model is written in Python as a free and open-source framework. This new model is then employed to generate data for deep learning (i.e., training and testing) and utilized as the independent testing environment to validate predictions for controlling groundwater. Integration with other deep learning algorithms is an advantage of the proposed model compared with the commercially available software for transient seepage analysis. Additionally, the governing equation and the auxiliary equations applied in this study can be easily modified to allow for more complex seepage analysis.

The rest of the paper is organized as follows: Section 2 discusses the data generation approaches; Section 3 introduces the background of LSTM, then explains the proposed model, required data preprocessing, and evaluation criteria; Section 4 presents the results and discussions; and, finally, Section 5 concludes the study.

2. Data Acquisition

To generate the data for training and testing of the proposed LSTM model, a transient seepage model was developed to simulate a lab-scale geosystem equipped with three pumps and subjected to rainfall events. In our previous study, we established a similar physical model that seamlessly coupled seepage and slope stability analysis to better understand the influence of water level changes on the safety status of slopes [33]. The computational framework of the physical model was cross-validated by commercial software in our previous study [33].

The goal of this study is to explore integrating the physical geosystems and the man-made interventions with a deep learning algorithm (i.e., LSTM) to learn the policy of water pumping from historical data and to establish an autonomous water pumping system for long-term use. Landslides are often the results of extreme weather events, as explained in Section 1. The data directly influencing slope stability for landslide prevention include the following four categories: (1) the geomaterial characteristics of the slope, mainly the associated unsaturated and saturated soil parameters; (2) the geometry of the slope and the pumps; (3) rainfall data and detailed information on pumps including each pump's flow rate policy, location, and initial water table; and (4) time variables. Thus, we first present the data generation in Section 2.

2.1. Generation of Rainfall Data

The design of rainfall events requires the determination of three basic parameters: (1) duration, (2) depth of precipitation, and (3) rain intensity. According to German guidelines, DVWK (Deutschen Verbandes für Wasserwirtschaft und Kulturbau (DVWK)), there are four possible intensity distribution patterns for rainfall events, as shown in Figure 1 [34]. In the first typical rainfall pattern, Figure 1a, the rainfall intensity is constant over time. Figure 1b shows the rainfall distribution with a descending pattern. In this type of distribution, maximum rainfall intensity occurs at the onset of precipitation. In the third type of distribution shown in Figure 1c, the maximum intensity occurs in the middle of the rainfall event duration, similar to a normal distribution. In the last type in Figure 1d, the rainfall intensity increases over time and reaches the maximum intensity at the end of the precipitation.

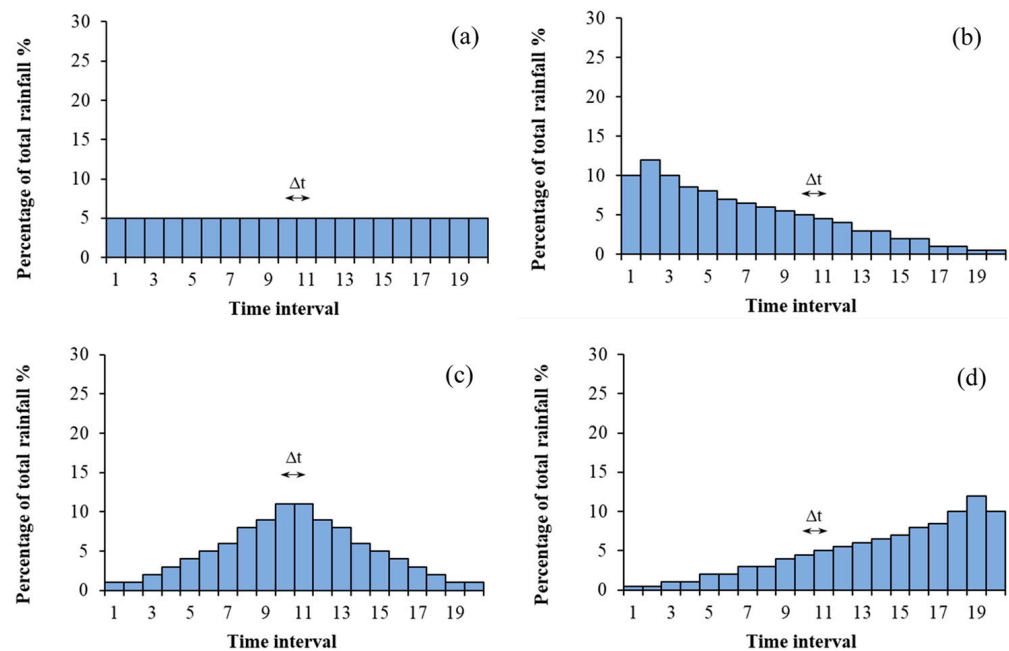


Figure 1. Typical rainfall intensity distributions with (a) constant intensity over time, (b) maximum intensity at the beginning of precipitation, (c) maximum intensity in the middle of precipitation, and (d) maximum intensity at the end of precipitation.

Forty-eight sets of rainfall data with a wide range of varieties in duration, total precipitation depth, and patterns were generated. Three values, (i.e., 10 min, 15 min, and 20 min), were selected for the rainfall duration. Four values were considered for the total rainfall depth: 10 mm, 15 mm, 20 mm, and 25 mm. Generated rainfall events typically have one of the four typical patterns described in Figure 1a–d, which will be referred to as types “a”, “b”, “c”, and “d”, respectively. It is noted that the maximum rainfall intensity in all cases was less than the saturated hydraulic conductivity of the chosen soil to prevent runoff, allowing all precipitation to infiltrate the soil [35].

2.2. Pump Flow Rate Data Acquired from Transient Seepage Analysis

2.2.1. Governing Equation for the Transient Seepage Model

A transient seepage analysis was performed to obtain the corresponding pump’s flow rate for each rainfall event. The governing equation for a transient saturated-unsaturated seepage model was obtained by modifying the Richards Equation [36],

$$S \frac{\partial(h+z)}{\partial t} = K \times \nabla(\nabla(h+z)) + q_H, \quad (1)$$

where h is the pressure head with the unit of meter, z is the elevation head with the unit of meter, q_H is the sink/source term representing the applied flux at the boundaries with the unit of meter per second, and S and K are defined based on the saturation degree as follows,

$$\begin{cases} \text{Saturated Flow} \rightarrow K = K_s, S = S_s \\ \text{Unsaturated Flow} \rightarrow K = K_s K_r, S = S_c \end{cases} \quad (2)$$

where, for the saturated flow, S_s is the specific storage of saturated flow with the unit of 1/m, and K_s is the saturated hydraulic conductivity with the unit of meter per second; for the unsaturated flow, S_c is the specific moisture content with the unit of 1/m and $K_s K_r$ is the unsaturated hydraulic conductivity, in which K_r is the relative hydraulic conductivity. Relative hydraulic conductivity defines the way in which the hydraulic conductivity

changes with the degree of effective saturation (S_e). The following equation proposed by van Genuchten [37] was adopted for formulating K_r :

$$K_r = S_e^b \left(1 - \left(1 - S_e^{1/a} \right)^a \right)^2, \tag{3}$$

$$S_e = \left[1 + \left(\frac{\psi}{P_0} \right)^{\frac{1}{1-a}} \right]^{-a}, \tag{4}$$

where a , b , and P_0 (with the unit of Pascal) are fitting parameters derived from the Soil-Water Characteristic Curve (SWCC), S_e is the effective saturation degree, and ψ (with the unit of Pascal) is the matric suction [17]. The matric suction is calculated using $\psi = \gamma_w h$, where γ_w is the unit weight of water.

The specific moisture content for unsaturated flow (S_c) defines the rate of change in the water content per unit change of the negative water head [36]. S_c is the derivative of volumetric water content θ with respect to h :

$$S_c = \left| \frac{\partial \theta}{\partial h} \right| = n \frac{\partial S_e}{\partial h} = \frac{na}{h(a-1)} \left(1 + \left(\frac{\psi}{P_0} \right)^{\frac{1}{1-a}} \right)^{-a-1} \left(\frac{\psi}{P_0} \right)^{\frac{1}{1-a}}, \tag{5}$$

where n is the soil porosity.

Specific storage of the saturated flow S_s is the volume of water released from a unit volume of aquifer per unit decline in the hydraulic head. S_s is a function of the compressibility values of soil and water and the soil porosity [38]:

$$S_s = \frac{1}{V_t} \frac{dV_w}{dh} = \rho_w g (C_s + nC_w), \tag{6}$$

where ρ_w is the density of water with the SI unit of kg/m^3 , g is the gravitational acceleration, C_s is the soil compressibility with the SI unit of ms^2/kg , and C_w is the water compressibility with the SI unit of ms^2/kg .

2.2.2. Geometry and Boundary Conditions

As shown in Figure 2, a typical slope profile consisting of three pumps, was used in the analyses. The slope geometry, boundary conditions, and location of pumps were determined from the lab-dimensional model. It was assumed that there was no input flux other than rainfall. Pumps were modeled as a sinkhole. The diameter of the sinkhole for all of the three pumps was 2 inches. The initial groundwater table is set to the level of 6 inches.

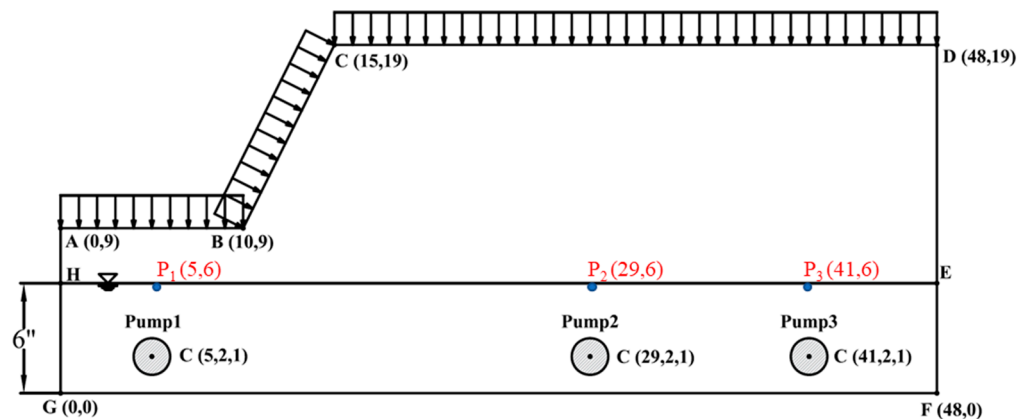


Figure 2. Cross-section of soil slope equipped with pumps subjected to rainfall (Units: Inch).

The no-flux boundary condition was applied to the left and right sides of the slope above the groundwater table (i.e., DF and GA) and along the bottom of the slope (i.e., FG). This type of boundary condition was formulated using a Neumann boundary condition as Equation (7),

$$\text{Neumann BC : } -\nabla(h+z) \cdot \vec{n} = 0 \text{ on } \Gamma_{(DF, FG, GA)} \text{ for } t > 0, \quad (7)$$

The slope's surface boundaries (i.e., AB, BC, and CD) were set to the influx boundary conditions to simulate rainfall infiltration at the slope's surface. Water ponding was not considered for the slope's surface since the rain intensities (I_r) were less than the soil infiltration capacity,

$$\text{Neumann BC : } -\nabla(h+z) \cdot \vec{n} = I_r \text{ on } \Gamma_{(AB, BC, CD)} \text{ for } t > 0, \quad (8)$$

The outflux boundary condition was adopted for the sinkhole boundaries of the pumps. The definition of flux is the discharge per unit area per unit time (i.e., m/s). Based on this definition, the Neumann boundary condition for the pumps was formulated as Equation (9):

$$\text{Neumann BC : } -\nabla(h+z) \cdot \vec{n} = \alpha \frac{Q_p}{2\pi r} \text{ on } \Gamma_{(\text{Pump 1, Pump 2, Pump 3})} \text{ for } t > 0, \quad (9)$$

where Q_p is the full capacity of the pumps, which was considered to be 1 gpm (gallon per minute), r is the radius of the sinkhole for the pumps, and α is a value between 0 and 1 that was defined based on a simple policy for each pump to keep the water level at the target level. The target level in this model is the initial water level (i.e., 6"). The policies for pumps 1, 2, and 3 were formulated according to the range of water head at points P_1 , P_2 , and P_3 (see Figure 2), which are presented in Equations (10)–(12), respectively:

$$\begin{aligned} & \text{If } h(P_1) = 0 \text{ m} \rightarrow \alpha = 0 \\ \text{Pump 1 : } & \text{If } 0 < h(P_1) < 0.4 \text{ m} \rightarrow \alpha = h(P_1) \times 2.5, \\ & \text{If } h(P_1) \geq 0.4 \text{ m} \rightarrow \alpha = 1 \end{aligned} \quad (10)$$

$$\begin{aligned} & \text{If } h(P_2) = 0 \text{ m} \rightarrow \alpha = 0 \\ \text{Pump 2 : } & \text{If } 0 < h(P_2) < 0.1 \text{ m} \rightarrow \alpha = h(P_2) \times 10, \\ & \text{If } h(P_2) \geq 0.1 \text{ m} \rightarrow \alpha = 1 \end{aligned} \quad (11)$$

$$\begin{aligned} & \text{If } h(P_3) = 0 \text{ m} \rightarrow \alpha = 0 \\ \text{Pump 3 : } & \text{If } 0 < h(P_3) < 0.15 \text{ m} \rightarrow \alpha = h(P_3) \times 6.67, \\ & \text{If } h(P_3) \geq 0.15 \text{ m} \rightarrow \alpha = 1 \end{aligned} \quad (12)$$

where $h(P_1)$, $h(P_2)$, and $h(P_3)$ are the water head values at points P_1 , P_2 , and P_3 , respectively. The purpose of this study is to prove that a control policy such as that used by an experienced engineer to control the drainage of a geosystem can be learned. If successful, we can utilize existing control data or engineers' experiences to generate an autonomous subsurface drainage control system. To support proof of concept, the above policy and a clear mathematical formulation were adopted.

2.2.3. Soil Properties

Soil properties for the transient saturated-unsaturated flow model are presented in Table 1. Figure 3 represents the SWCC and the relationship between the relative hydraulic conductivity and the effective saturation for the chosen sandy soil. It is noted that the values for the specific storage (S_s) for saturated flow were assumed based on the ranges provided by [38] for sandy soils. To focus on the proof of the concept in this study, sandy soil was selected in this study, considering the relatively short seepage analysis time and low computational cost. Proving this concept with coarse-grained soils can also confirm

the applicability of this method to other soils as long as the governing physics principles behind the transient seepage analysis do not change significantly.

Table 1. Soil properties for the transient saturated-unsaturated flow model.

Model Input	Definition	Sand
K_s	Saturated hydraulic conductivity (m/s)	3×10^{-4}
S_s	Saturated specific storage (1/m)	1×10^{-4}
a	Empirical parameter	0.6
b	Empirical parameter	0.5
P_0	Empirical parameter (Pa)	1200
n	Porosity	0.32

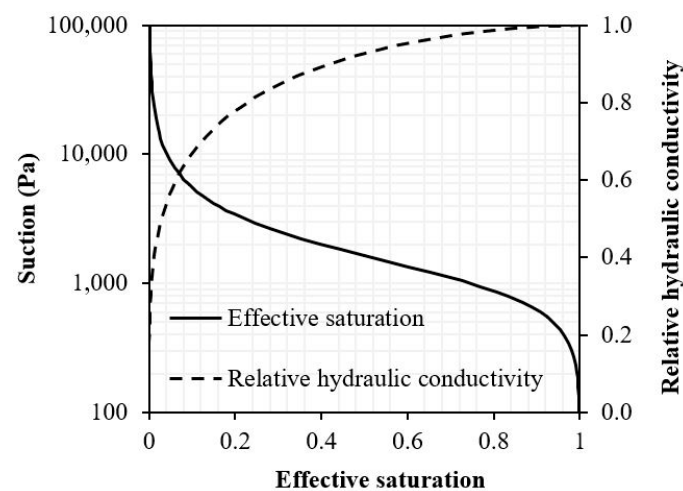


Figure 3. SWCC and relative hydraulic conductivity K_r , of the soil.

2.2.4. Numerical Implementation

The seepage analysis was implemented using the DOLFIN package, a Python interface of FEniCS. FEniCS is a finite element analysis platform to solve nonlinear partial differential equations (PDEs). The procedure for solving the governing equation for the transient seepage analysis was summarized in Figure 4. The required input parameters to solve the PDE included unsaturated soil characteristics (a , b , P_0 , n); saturated specific storage (S_s); saturated hydraulic conductivity (K_s); slope geometry; the geometry of pumps; the location of points P_1 , P_2 , and P_3 , boundary conditions including the rain intensity and the pumps' flow rate policy, the initial groundwater table, and time variables (total time $T = 120$ min, and time interval $\Delta t = 1$ min). A total time of 120 min was considered in the seepage analysis to ensure that the groundwater reaches its initial condition at the end of pumping. A mesh of 3-node Lagrangian elements was generated in the computational domain of the slope (Ω). Boundary conditions were applied to the subdomain as described in Section 2.2.2. The initial boundary condition was specified as the solution to the PDE at $t = 0$. Auxiliary equations for the unsaturated hydraulic conductivity and effective saturation were defined. Next, the PDE for the governing equation was reformulated as a finite element variational problem. For each time step, the boundary conditions for the slope's surface and the pumps were updated based on the rain intensity and water head of points P_1 , P_2 , and P_3 at the time t_i . Solving the PDE gives the total water head distribution at the time t_i , which was utilized to update the pumps' flow rate at the time t_{i+1} . The output of the seepage analysis is the flow rate of pumps during and after the rainfall event.

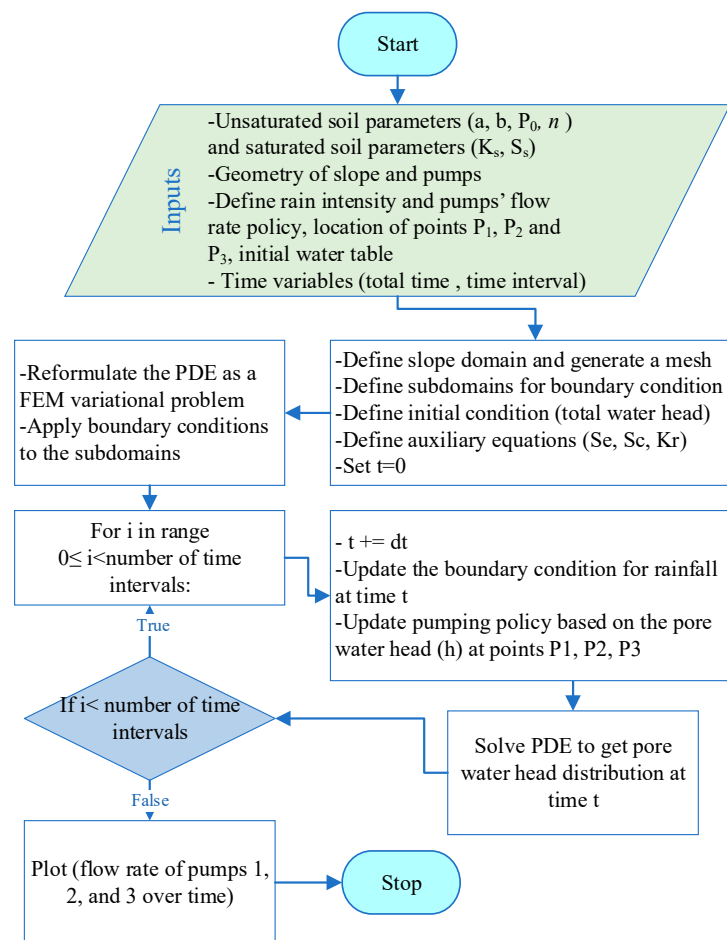


Figure 4. Flowchart of the written python code using DOLFIN package to solve the governing equation for transient seepage analysis.

Figure 5 shows the output of the seepage analysis for pumps’ flow rate in a typical rainfall event. The seepage analysis was performed for all generated rainfall events (48 sets). Since the flow rates of all the three pumps reached zero at the end of each set, the datasets from all 48 sets were combined to produce a time series of the pumps’ flow rate and rainfall. The time series of rainfall shown in Figure 6 was used as the input to the LSTM model, and the corresponding flow rates for pumps 1, 2, and 3 shown in Figure 7 were defined as the output.

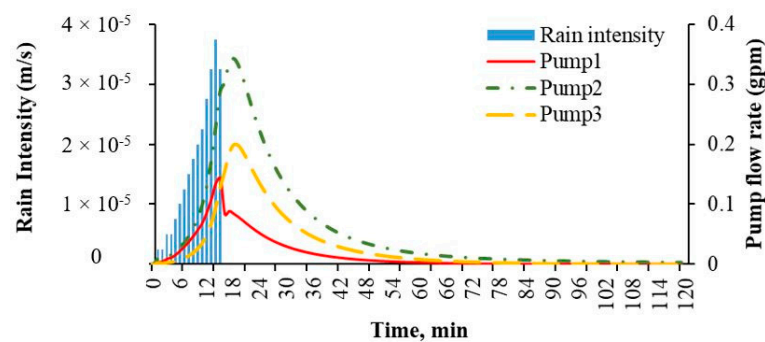


Figure 5. Pump flow rates as the output of the seepage analysis in a typical generated rainfall event (duration = 15 min, rainfall depth = 15 mm, and pattern “d”).

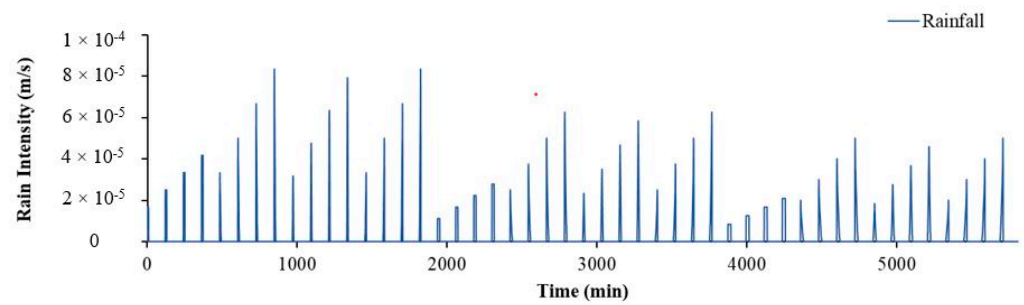


Figure 6. Generated time series of rainfall.

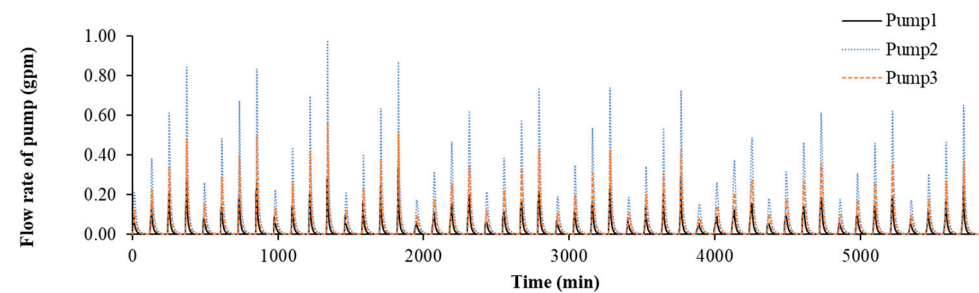


Figure 7. Time series of flow rates for pumps 1, 2, and 3 corresponding to the generated rainfall data.

3. Methodology

In the past decades, machine learning techniques have been applied to autonomous systems because of their ability to discover knowledge from observed data [39]. These techniques have been widely used because they do not require predefined input/output relationships [40]. The LSTM is a type of recurrent neural network (RNN) that has a memory cell to store information and dependencies long term [41]. The LSTM has shown promising results in time-series prediction tasks such as traffic flow prediction [42], travel time prediction [43], predicting water table depth in an agricultural area [44], flood forecasting [45], rainfall prediction [46], prediction of water levels in sewer systems [47], steam flow forecasting of small rivers [48], and prediction of displacement in multifactor-induced landslides [49]. Besides, the LSTM performs better when capturing various data patterns than the traditional neural network models [50,51]. Thus, the deep learning model for the proposed autonomous water pumping system was established with the LSTM.

3.1. Background: Long Short-Term Memory

The LSTM was initially proposed by [41]. The LSTM is a special kind of RNN that transfers historical information and long-term dependencies through time sequences [52]. Figure 8 shows the general structure of an RNN, which includes three layers: the input, hidden, and output layers. The output of each time step depends on the current time step's input and the output from the previous time step. This dependency is transferred by the hidden layer units connected through time. In an RNN unit, the recurrent hidden state at time step t (i.e., h_t) is updated using Equation (13).

$$h_t = \tanh(U \cdot X_t + W \cdot h_{t-1} + b_h), \quad (13)$$

where X_t is the input at time-step t , U is the weight matrix between the input and the hidden layer, W is the weight matrix between hidden units, h_{t-1} is the hidden state in the previous time-step, b_h is the bias vector of the hidden layer, and \tanh is a hyperbolic tangent function that transforms the input to a value between -1 and 1 . The output of the time step t (i.e., y_t) is obtained from Equation (14).

$$y_t = V \cdot h_t + b_y, \quad (14)$$

where V is the weight matrix between the hidden layer and the output layer and b_y is the bias vector of the output layer. Although the RNN is a rigorous tool in time series prediction, it could not preserve the dependencies for a long time [52,53].

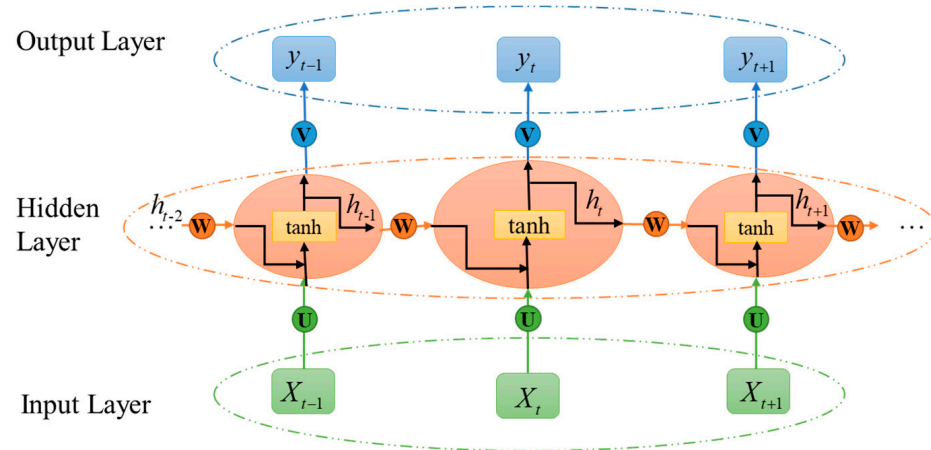


Figure 8. The general structure of the recurrent neural network (RNN).

The LSTM has a memory cell that helps maintain long-term dependencies, as illustrated in Figure 9. Three gates control this memory cell: input gate, forget gate, and output gate [43,54]. The state of the three gates is updated in each time step using the following equations Equations (15)–(17).

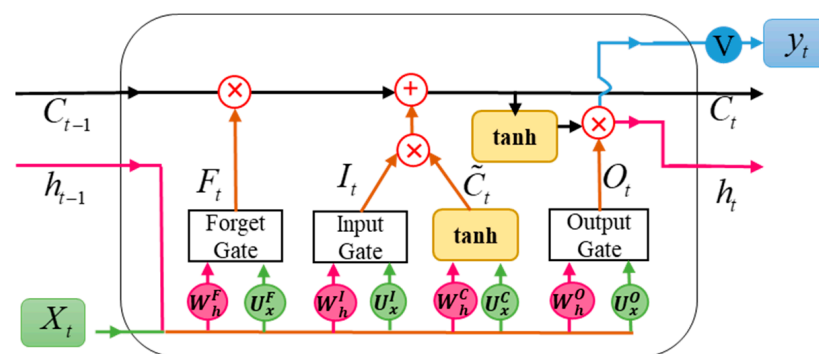


Figure 9. The structure of the LSTM unit.

The updated state of the input gate at the time t :

$$I_t = \sigma(U_X^I \cdot X_t + W_h^I \cdot h_{t-1} + b_I), \tag{15}$$

The updated state of the forget gate at the time t :

$$F_t = \sigma(U_X^F \cdot X_t + W_h^F \cdot h_{t-1} + b_F), \tag{16}$$

The updated state of the output gate at the time t :

$$O_t = \sigma(U_X^O \cdot X_t + W_h^O \cdot h_{t-1} + b_O), \tag{17}$$

where U_X^I , U_X^F , and U_X^O are the weight matrices of the input, forget, and output gates for the input (X_t), respectively; W_h^I , W_h^F , and W_h^O are the weight matrices of the three gates for h_{t-1} in which h_{t-1} is the hidden state in the previous time-step, $t - 1$; b_I , b_F , and b_O are the bias vectors of the three gates; σ is the sigmoid activation function, which transforms the input to a value between 0 and 1 [43].

The memory cell is updated each time step by the forget gate and the input gate, enabling the cell to store information for a long time without a gradient vanishing problem. The output of the memory cell C_t is calculated using Equation (18). The forget gate is responsible for forgetting irrelevant information from the past [52,54].

$$C_t = (I_t \otimes \tilde{C}_t) + (F_t \otimes C_{t-1}), \quad (18)$$

where C_{t-1} is the previous state of the memory cell and \tilde{C}_t is the candidate input for the memory cell at a time t :

$$\tilde{C}_t = \tanh(U_X^C \cdot X_t + W_h^C \cdot h_{t-1} + b_C), \quad (19)$$

where U_X^C and W_h^C are the weight matrices of the memory cell for X_t and h_{t-1} , respectively, and b_C is the bias vector of the memory cell [43].

The hidden layer's state at time t (h_t) is calculated with Equation (20):

$$h_t = O_t \otimes \tanh(C_t). \quad (20)$$

The output of the LSTM unit is obtained with Equation (21):

$$y_t = V \cdot h_t + b_y, \quad (21)$$

where V is the weight matrix between the hidden layer and the output layer and b_y is the bias vector of the output layer [43].

3.2. Proposed LSTM Model

In the practical water pumping system for rainfall-induced landslide prevention, the flow rate of pumps is the operating parameter that workers can manually control. Thus, this study aims to predict the corresponding flow rate of pumps for the given rainfall data to realize the autonomous water pumping system and provide timely interventions. The pumps' flow rates depend on the information given for the rainfall intensity information provided. The LSTM network was constructed using the Keras deep learning library (Version 2.2.4) installed with the Tensorflow backend. It is a common practice in the literature to divide the total data into the training and testing sets, with a ratio of 70–80 percent for training and 20–30 percent for testing [55–57]. The data for 36 sets of rainfall events ($36 \times 121 = 4356$ rows), including different patterns for the rainfalls and the corresponding data for the pumps' flow rates (75% of the dataset), were used to train the proposed LSTM model. Then, the model's performance on the remaining 25% of the dataset, including 12 sets of rainfall events ($12 \times 121 = 1452$ rows) with a new pattern and the corresponding data for the pumps' flow rates, was validated. Since this problem is a regression task, the mean absolute error (MAE) was used as a loss function and is calculated using Equation (22):

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|, \quad (22)$$

where n is the total number of data, y_i is the actual value, and \hat{y}_i is the predicted value for the pump's flow rate.

3.3. Data Preprocessing

3.3.1. Scaling

Datasets including variables in different ranges and units might harm the learning ability of the model [44]. Scaling the dataset can improve the model's performance in the

training process [58]. The Min-Max normalization method was applied to each feature separately, which scaled the data values linearly between [0, 1] as follows:

$$X_{normalized} = \frac{x - x_{min}}{x_{max} - x_{min}}, \quad (23)$$

where x_{min} and x_{max} are the minimum and maximum values of features on their current scale [59].

3.3.2. Transform the Dataset into a Supervised Learning Problem

This step is required to use the LSTM algorithm. The purpose is to train the LSTM model to learn the dependencies in the dataset and map the output to the given input. To frame the dataset as a supervised learning problem, the input and the output of the model were defined for each time step. In this study, the input is the rainfall intensity in the next 120 min ($I_{r_t}, I_{r_{t+1}}, \dots, I_{r_{t+120}}$) and the expected output is the flow rates of pump 1 (i.e., $q_{1_t}, q_{1_{t+1}}, \dots, q_{1_{t+120}}$), pump 2 (i.e., $q_{2_t}, q_{2_{t+1}}, \dots, q_{2_{t+120}}$), and pump 3 (i.e., $q_{3_t}, q_{3_{t+1}}, \dots, q_{3_{t+120}}$) in the next 120 min.

3.4. Model Evaluation Metrics

The predictions were evaluated using three performance metrics, the root mean square error (RMSE), the mean absolute error (MAE), and the coefficient of determination (R^2). The RMSE and MAE are useful evaluation metrics for regression problems. The RMSE (Equation (24)) and MAE (Equation (22)) take a value between $[0, +\infty]$ and have the same unit as the variable of interest. Values of RMSE and MAE close to 0 indicate a model's good performance in prediction [60]. The R^2 -score, also called goodness of fit, measures how well a model can predict the desired variable and fit the observed data [61]. The R^2 (Equation (25)) gives a unitless score between 0 and 1 in which the score of 1 represents the optimal forecast model [62].

$$RMSE = \left[\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \right]^{\frac{1}{2}}, \quad (24)$$

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2}, \quad (25)$$

where n is the total number of data and \bar{y} is the mean value of the observed data.

4. Results and Discussions

In this section, the results generated in the search for the optimal LSTM architecture are presented. Then, the predictions obtained from the optimal architecture for the testing set are validated. Finally, the influence of rainfall patterns, which could affect the generalization of the method, is discussed.

4.1. Search for Optimal LSTM Architecture and Hyperparameters

The optimal LSTM architecture was achieved by optimizing the LSTM hyperparameters. LSTM hyperparameters include the numbers of LSTM layers, units in each LSTM layer, fully connected layers (or dense layers), units in each dense layer, and epochs, as well as the batch size and the optimization learning algorithm. The number of epochs defines the number of times the algorithm completes the learning process on the entire dataset (training and testing sets). The dataset is usually divided into a number of batches to complete one epoch. The batch size refers to the number of training samples used in a single batch. By optimizing the results of the evaluation metrics, the number of epochs and the batch size for all models were determined to be 300 and 50, respectively. It is noted that

each LSTM model was trained using an adaptive optimization learning algorithm, (i.e., Adam), with a learning rate of 0.001.

Hyperparameters were chosen from a set of selected candidate values, as shown in Table 2. In the experimented models, the number of LSTM layers was set to be 1, 2, or 3 layers, and the number of units for an LSTM layer were selected from values of 100, 50, 30, and 20. The models have one or two dense layers. In the sequence of dense layers, the number of units for the last dense layer represents the number of outputs, or predictions. In the case of two dense layers, 500 was chosen as the number of units was chosen to be 500 and 363 was selected for the first and second dense layers, respectively. In the case of one dense layer, the number of units was set the same as the output dimension of 363. The output of this model includes 120 min (the next 121 time steps) of flow rates for three pumps ($3 \times 121 = 363$ data points).

Table 2. Selected candidate values for hyperparameters of the LSTM model.

Hyperparameter	Candidate Values
Number of LSTM layers	1, 2, 3
LSTM unit sizes	100, 50, 30, 20
Number of dense layers	1, 2
Dense unit sizes	500, 363

Models with the hyperparameters mentioned above were evaluated using performance metrics of RMSE and R^2 as presented in Table 3. The optimal hyperparameters for the proposed model were achieved by evaluating the RMSE and R^2 scores. It can be seen from Table 3 that increasing the LSTM layers to two or three did not improve the RMSE or R^2 scores. Thus, the number of LSTM layers was optimized to one. For the unit size of the LSTM layer, it was found that a unit number smaller than 50 affected the learning ability of the model. For 20 and 30 units, the recorded R^2 scores are 0.949 and 0.932, respectively, which are smaller than $R^2 = 0.962$ for 50 units. A higher number of units for the LSTM layer (i.e., 100) did not improve the model's accuracy. Based on the evaluation results, 50 was determined the optimal value for the number of units of the LSTM layer. With an R^2 score of 0.962, the model with only one dense layer showed better performance than two dense layers which had an R^2 score of 0.952.

Table 3. Experimented LSTM models with selected hyperparameters.

Number of LSTM Layers	Number of Units for LSTM Layers	Number of Fully Connected Layers (Dense)	Number of Units for Dense Layers	Batch Size	Epochs	Optimization Algorithm, Learning Rate	RMSE (GPM)	R^2
1	100	1	363	50	300	Adam, 0.001	0.017	0.954
1	50	1	363	50	300	Adam, 0.001	0.015	0.962
1	30	1	363	50	300	Adam, 0.001	0.018	0.946
1	20	1	363	50	300	Adam, 0.001	0.018	0.945
1	50	2	500, 363	50	300	Adam, 0.001	0.017	0.952
2	50, 50	1	363	50	300	Adam, 0.001	0.018	0.950
2	50, 50	2	500, 363	50	300	Adam, 0.001	0.018	0.946
3	50, 50, 50	1	363	50	300	Adam, 0.001	0.019	0.943
3	50, 50, 50	2	500, 363	50	300	Adam, 0.001	0.019	0.943

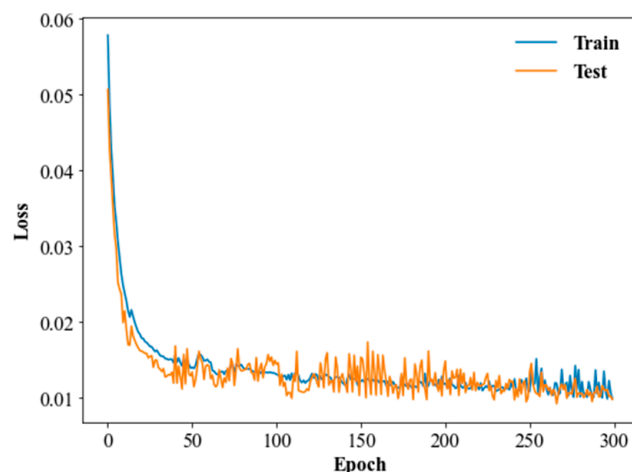
A summary of the optimized hyperparameters is reported in Table 4. The optimal architecture comprises one LSTM layer with 50 units and a fully connected layer (or dense layer) at the top of the LSTM layer. The number of units for the dense layer is 363, the same as the output size. The epoch and the batch size selected for this model were 300 and 50, respectively. The proposed LSTM model has the best performance in the evaluation metrics: RMSE = 0.015 GPM and $R^2 = 0.962$.

Table 4. Proposed architecture for the optimal LSTM model.

Model Hyperparameters	Optimal Value
Number of the LSTM layers	1
LSTM unit size	50
Number of the dense layers	1
Dense unit size	363
Epochs	300
Batch size	50
Optimization algorithm, learning rate	Adam, 0.001

4.2. Training and Testing with the Optimal LSTM Architecture

The model was trained with the optimal architecture introduced in Table 4, and the trained model was then validated on the testing set. Figure 10 shows the learning curves of the training and the testing sets for the optimal LSTM architecture. The MAE score, which was used as a loss function for both training and testing sets, decreases to the point of stability as the number of epochs increases. Additionally, there is a small gap between the training and testing loss curves. Meeting these two criteria indicates a good fit for the model.

**Figure 10.** Loss of the training and testing sets for the optimal LSTM architecture.

The predictions of flow rates for pumps for the testing sets with the trained model were validated using the results of seepage analysis. Figure 11a–c compares the predicted flow rates from the proposed LSTM model with the values from the seepage analysis for pumps 1, 2, and 3, respectively. The input for both the LSTM model and the seepage analysis is the rainfall data, and the output is the pump's flow rate. In the seepage analysis, the pump flow rate was obtained based on the predefined pumping policy for each pump, while the proposed LSTM model could learn the hidden pumping policies for each pump within the training dataset. Figure 11a–c supports the idea that the trained model can predict the required pumps' flow rate for the testing set to keep the water table at the target level in response to the rainfall events. The proposed LSTM model can closely map the input data (i.e., set of rainfall data) to the expected output data (i.e., corresponding pumps' flow rate) without requiring conducting a seepage analysis with prescribed pumping policies. An evaluation of the predictions, which is presented in Table 5, also confirmed this finding.

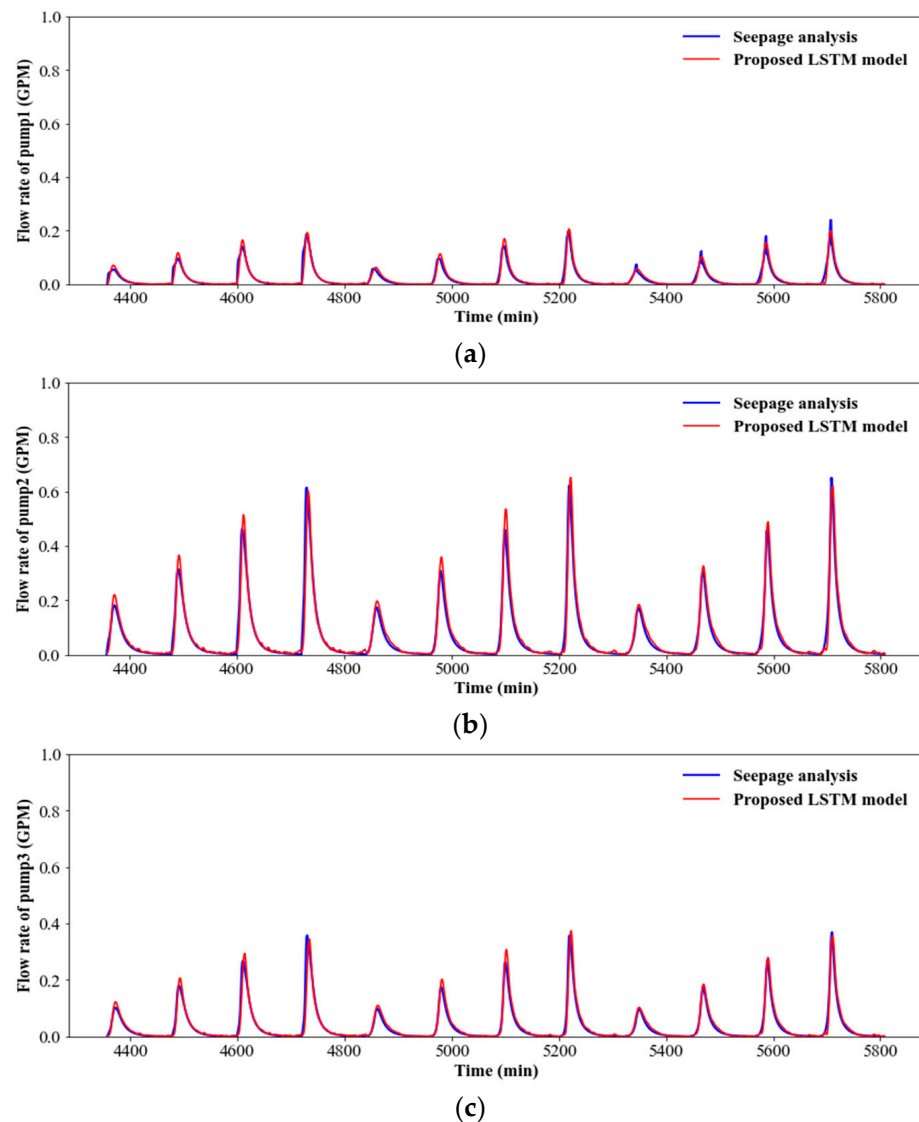


Figure 11. Comparison of pumps' flow rate from the seepage analysis and predicted values using the LSTM model for (a) pump 1, (b) pump 2, and (c) pump 3.

Table 5. Performance metrics of the proposed model on the testing set.

Predictions	RMSE (GPM)	MAE (GPM)	R ²
Pump 1	0.007	0.003	0.958
Pump 2	0.021	0.009	0.962
Pump 3	0.014	0.006	0.954

From the proposed model, predictions regarding the flow rate of pumps were evaluated using the performance metrics of R², RMSE, and MAE. Regardless of the output's unit, values of R² close to 1 represent high prediction accuracy. The R² score of the flow rate predictions for pumps 1, 2, and 3 are 0.958, 0.962, and 0.954, respectively. The achieved high values of the R² score indicate how well the proposed LSTM model fits the observed data from the seepage analysis.

Unlike the R², RMSE and MAE carry the same unit as the predicted values for the pump flow rates in this study. The RMSE is the square root of the average squared differences between the predicted values from the proposed LSTM model and the actual values from the seepage analysis. The RMSE of the flow rate predictions for pumps 1, 2, and 3 are 0.007, 0.021, and 0.014, respectively. The RMSE values close to 0 represent

a better prediction. Also, taking the range of a pump's flow rate into consideration can help interpret the RMSE values. The range of observed flow rates of pumps 1, 2, and 3 from the seepage analysis in the testing set is 0–0.24 GPM, 0–0.65 GPM, and 0–0.37 GPM, respectively. Thus, the RMSE of the predicted flow rates for the three pumps is relatively small compared with the range of changes in the flow rates of the pumps.

The MAE is the average of differences between the predicted values from the proposed LSTM model and the actual values from the seepage analysis. The MAE of the flow rate predictions for pumps 1, 2, and 3 are 0.003, 0.009, and 0.006, respectively. Values of MAE are very small, which means the model can predict the pump's flow rate with high accuracy. The evaluation of the predictions using various measurements verifies the promising performance of the proposed LSTM model in learning the pumping policy and predicting a pump's flow rate.

4.3. Discussion 1: Application of the LSTM Model in Controlling the Groundwater

This subsection discusses different approaches for employing the proposed LSTM model in controlling groundwater level as an intervention method for the geotechnical CPSs studied in this paper. For this purpose, the predicted flow rates from the LSTM model were applied to the pumps in the slope to control the groundwater table. A seepage analysis was then conducted to obtain the groundwater levels at the three points of P_1 , P_2 , and P_3 during the rainfall events.

The pump flow rates for the 12 events in the testing set can be continuously forecasted using the LSTM model trained with 36 events. In the method discussed in previous sections, any error present in the initial prediction will be carried forward to subsequent predictions. While this method is flexible and easy to implement, the accuracy of predictions will be reduced in later predictions. To reduce accumulated errors, pump flow rates for each rainfall event can be predicted independently. In this method, a separate LSTM model with the optimal architecture, introduced in Section 4.1, was created to forecast the pump flow rate for each event. After forecasting the pump flow rate for one event, the actual data for that event was added to the training set to reduce the accumulated error for the flow rate predictions of the next rainfall event. By moving forward, the size of the training set increased, which improved the model's performance on the next event's predictions.

There are two options for applying the predicted pump flow rate in the seepage analysis to control the groundwater. One option is to reset the groundwater table to the initial condition at the beginning of each rainfall event. While this option can help evaluate the pump flow rate for each rainfall event separately, it cannot represent the real-world application. Alternatively, the groundwater level reset could be omitted during the 12 rainfall events to simulate reality.

Based on the above descriptions, three approaches were investigated for employing the proposed LSTM model by combining the available options for predicting the flow rates of pumps and applying them in the seepage analysis to control the groundwater level. In the first approach, pump flow rates for 12 rainfall events in the testing set were continuously predicted using the LSTM model trained with 36 events. Then, the groundwater table was reset to the initial condition at the beginning of each event to apply the pump flow rate in the seepage analysis. In the second approach, pump flow rates were predicted similar to the first approach. Then, the predicted flow rates were continuously implemented in the seepage analysis without resetting the groundwater condition at the beginning of each event. In the third approach, the training set for the LSTM model was recalibrated after forecasting the pump flow rate for each rainfall event to reduce the accumulated error in predictions. Then, the predicted values were continuously applied to the pumps during the rainfall events to control groundwater level.

Figure 12a–c displays the groundwater level from the three approaches at points of P_1 , P_2 , and P_3 , respectively. To evaluate the application of the LSTM model, the groundwater levels from the three approaches were compared with the groundwater levels obtained

with the prescribed pumping policies corresponding to the rainfall events at the points mentioned above.

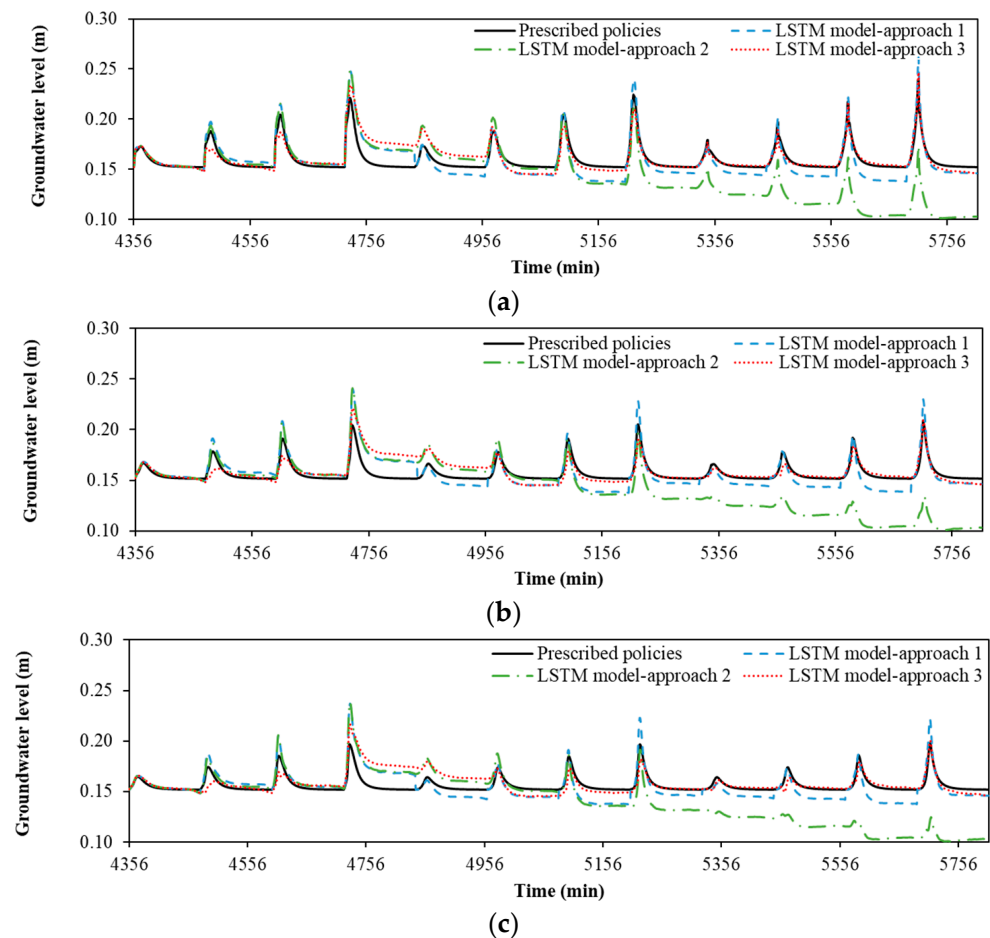


Figure 12. Comparison of the groundwater level using the three approaches of applying the proposed LSTM model and the predefined pumping policies at (a) point P_1 , (b) point P_2 , and (c) point P_3 .

A comparison of the groundwater levels using the prescribed pumping policies and approach 1 for employing the LSTM model shows that approach 1 has a discontinuity in control of the water table between the rainfall events. Despite positive results during the rainfall events at the three points of P_1 , P_2 , and P_3 , this approach cannot represent real conditions since the groundwater level was reset to the initial condition at the beginning of each event. An advantage of this approach is that the predictions for each event can be independently evaluated from the other events.

Another comparison of the groundwater levels using the prescribed pumping policies and approach 2 for employing the LSTM model provides insight into the accumulated error in the predictions. While this approach is easy to implement, the accuracy of results will be reduced in time. To improve the accuracy of results, approach 3 was suggested for employing the LSTM model. The results for approach 3 demonstrate this approach can boost the performance of the LSTM model in controlling the groundwater level at each of the three points (P_1 , P_2 , and P_3).

To better evaluate the proposed three approaches for employing the LSTM model, the differences between the groundwater level from each approach and the prescribed pumping policies are shown in Figure 13. The groundwater level using the prescribed pumping policies performs as a benchmark for evaluation. Figure 13a–c displays the differences at points P_1 , P_2 , and P_3 , respectively.

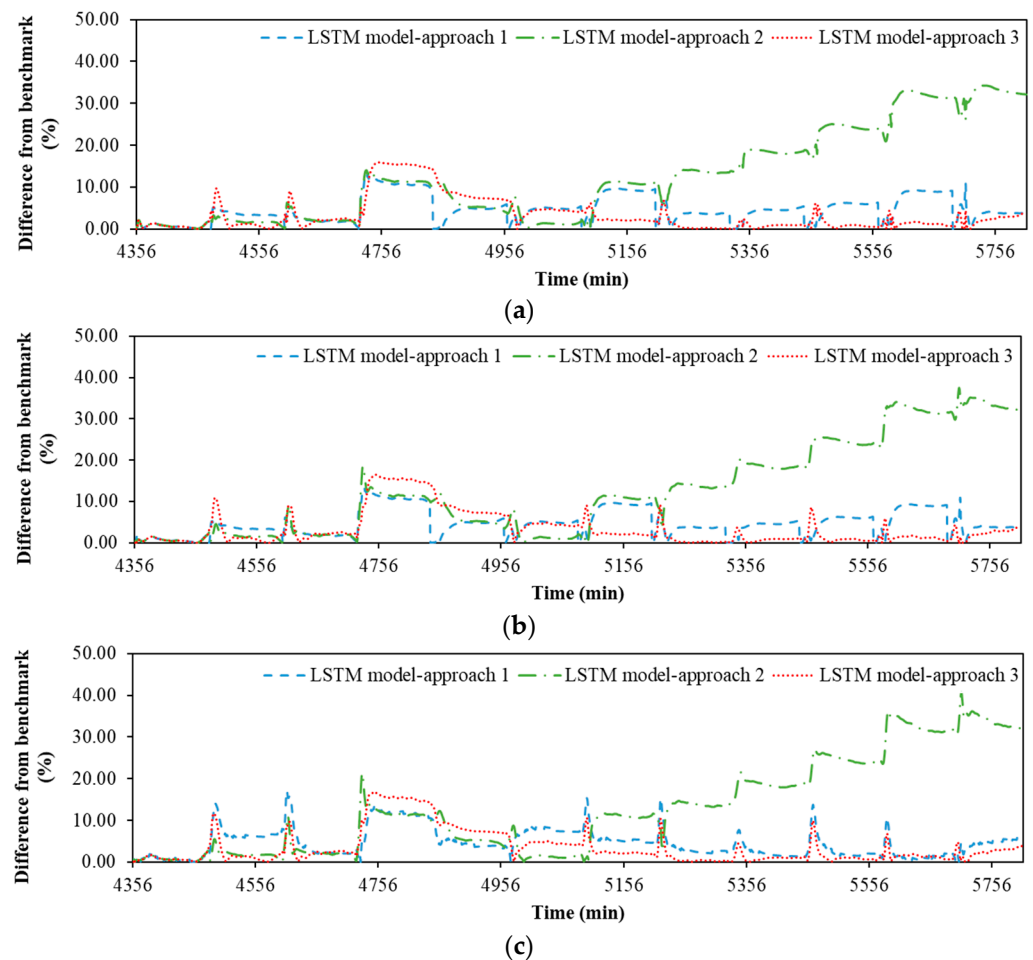


Figure 13. Differences of groundwater levels using the three approaches of employing the LSTM model from the groundwater level using the prescribed pumping policies (benchmark) at (a) point P_1 , (b) point P_2 , and (c) point P_3 .

The maximum difference from the benchmark for approach 1 at points of P_1 , P_2 , and P_3 are 13.3%, 16.0%, and 17.0%, respectively. This approach shows a difference of less than 10% in the groundwater level for most of the rainfall events. The average difference of the results for approach 1 from the benchmark is approximately 4.6% at all of the three points.

Approach 2 shows greater differences compared with the other two approaches. The differences from the benchmark for approach 2 at points P_1 , P_2 , and P_3 increase up to 34.4%, 37.5%, and 40.7%, respectively. The average difference of the results for this approach from the benchmark is approximately 12.7% at all of the three points. The results for approach 3 demonstrate noticeably lower values of differences from the benchmark. The average difference of the results for this approach from the benchmark is approximately 3.5% at all of the three points.

The above discussions conclude the LSTM model has an indisputable impact on controlling the groundwater table in the slope. The evaluation of the groundwater level from different approaches of applying the predictions indicates that the model's performance may decrease during long-sequential rainfall events. Thus, it is worthwhile to evaluate the predicted pump flow rates in controlling the groundwater table, in addition to the model evaluation metrics such as R^2 , RMSE, and MAE.

4.4. Discussion 2: Influence of Rainfall Patterns

This subsection is intended to evaluate the generalizability of the proposed LSTM model. A generalized model means the model is capable of performing well on an unseen

dataset. Since a small set of data with limited typical rainfall patterns was used in this study to train the LSTM model, it is beneficial to prove the model can be applied to real-world rainfall events with more complex patterns. Thus, the influence of the rainfall patterns on the accuracy of the model predictions was investigated. For this purpose, two rainfall datasets with different numbers of patterns were considered, as shown in Figure 14a,b. Both datasets include 24 rainfall events with the same variation in rainfall duration and depth. However, dataset 2 comprises more rainfall patterns types than dataset 1. The rainfall events in dataset 1 consist of only two types of patterns (i.e., types “a” and “d”), while the rainfall events in dataset 2 consist of four types of patterns (i.e., types “a”, “b”, “c”, and “d”). Figure 15a,b show the pumps’ flow rates for pumps 1, 2, and 3 corresponding to the rainfall data generated in Figure 14a,b, respectively.

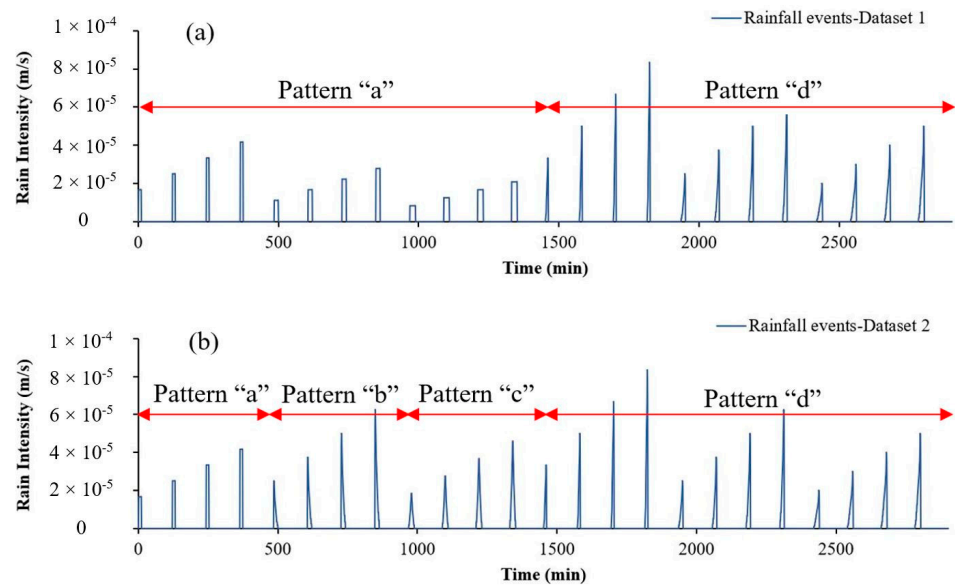


Figure 14. Two rainfall datasets with the different number of patterns (a) dataset 1 consists of rainfall events with pattern types of “a” and “d” and (b) dataset 2 consists of rainfall events with pattern types of “a”, “b”, “c”, and “d”.

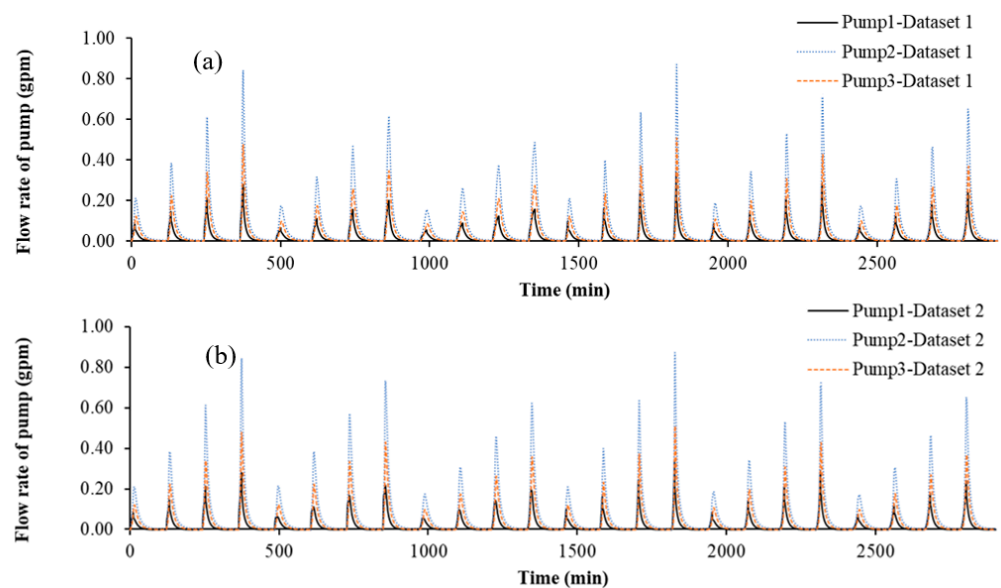


Figure 15. Time series of flow rate for pumps 1, 2, and 3 corresponding to the generated rainfall data in (a) dataset 1 shown in Figure 14a and (b) dataset 2 shown in Figure 14b.

An LSTM model was trained with the first 12 rainfall events and the corresponding pump flow rates for both datasets. Then, the model was validated with the next 12 rainfall events and their corresponding pump flow rates. The rainfall events in the testing set have the same type of pattern (“d”) in datasets 1 and 2.

Table 6 presents the evaluation results of the model trained with datasets 1 and 2. The evaluation metrics of RMSE and MAE for dataset 2 are lower than the values for dataset 1 in all three pumps. The R^2 score can better demonstrate the difference between the accuracy of the pump flow rate predictions for dataset 1 and dataset 2. The R^2 values for pumps 1, 2, and 3 in dataset 1 were 0.891, 0.922, and 0.910, respectively. By increasing the number of rainfall patterns for training the LSTM model from 1 in dataset 1 to 3 in dataset 2, the R^2 values for pumps 1, 2, and 3 were improved to 0.913, 0.935, and 0.925.

Table 6. Evaluation of the model trained with dataset 1 and dataset 2.

Predictions	RMSE (GPM)		MAE (GPM)		R^2	
	Dataset 1	Dataset 2	Dataset 1	Dataset 2	Dataset 1	Dataset 2
Pump 1	0.013	0.012	0.005	0.005	0.891	0.913
Pump 2	0.033	0.030	0.014	0.012	0.922	0.935
Pump 3	0.020	0.018	0.008	0.007	0.910	0.925

Additionally, comparing the comparison of the evaluation metrics for dataset 2 in Table 6 with the corresponding values in Table 5, it was found that the model trained with 36 rainfall events including the same types of patterns as dataset 2 has a better performance in predicting the pump’s flow rate. This means that the model trained with a dataset consisting of a larger number of rainfall events and pattern types improves the model’s performance in the prediction of the pump flow rates for a given new type of pattern.

4.5. Discussion 3: Limitation and Applicability of the Proposed Model

This subsection discusses the main limitations and application of the proposed LSTM model. They can be summarized as follows.

- Field conditions including more complex rainfall patterns and the pondering effect of precipitations were excluded from the numerical simulation for the data generation. Therefore, future studies can include actual measured rainfall data to improve the performance of the proposed LSTM model in capturing more complicated patterns.
- Ideal pump behaviors were assumed in the numerical simulation of the model, though the pump performance may vary depending on field conditions. Further experimental studies using the lab-scale geosystem described in Section 2.2.2 can help understand such limitations.
- This study employs a numerical simulation of a lab-scale geosystem for generating flow rate data to focus on the proof of the concept. With the same concept and framework, more complicated cases, such as field measurements from real-world slopes, can also be employed to train LSTM, as long as the core physics underlying the transient seepage analysis remains the same. Additional calibration and data processing may be required before applying the proposed model to data from field measurements.

5. Conclusions

This study is the first to investigate the ability of the LSTM deep learning algorithm to learn water pumping policy and the dependencies between the rain intensity and corresponding pump flow rates to establish a geotechnical CPS for rainfall-induced landslide prevention. The data for rainfall intensity was generated by a combination of different durations, depths, and patterns. The corresponding pump flow rate data was obtained by conducting the transient seepage analysis to satisfy desired drainage policies during rainfall events. The proposed LSTM model successfully predicted the multi-step ahead of the pump flow rate based on the rainfall intensity data given to the model as input. The major findings of this study are as follows:

- Evaluation metrics of RMSE, MAE, and R^2 showed a promising performance of the proposed LSTM model in predicting pump flow rates and learning the prescribed pumping policies. The R^2 values of 0.958, 0.962, and 0.954 for the pump flow rate predictions indicated high accuracy of the results.
- An assessment of the groundwater table after applying the pump flow rates demonstrated the model's performance could drop during long-sequential rainfall events. To avoid accumulated error in long-sequential rainfall events, it is suggested to use a separate LSTM model to predict the corresponding pump's flow rate for each rainfall event in the testing set.
- An evaluation of the influence of the rainfall patterns demonstrated that the performance of the proposed LSTM model can be improved by adding more and new rainfall patterns.
- As long as the nature (i.e., physics) underlying the transient seepage analysis is the same, the proposed method can be applied to real measurements of pump flow rates. It is noted that this may require additional calibration and data processing for the raw data.
- This study also presents a numerical framework written in Python to perform transient seepage analysis for a geosystem equipped with three pumps and subjected to rainfall events. Two main advantages of this framework are its integration with deep learning algorithms and its flexibility for considering more realistic field conditions.

Future studies can include more complex field conditions and rainfall patterns for generating training data, which can help to improve the performance of the proposed model.

Author Contributions: Conceptualization and methodology, Z.L. and A.B.; software and validation, A.B. and B.A.; writing—original draft preparation, A.B.; writing—review and editing, Y.S. and Z.L.; supervision and project administration, Z.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Science Foundation Grant No. 1742656 from the Geotechnical Engineering and Materials Program (now part of CMMI ECI). This work also used the Extreme Science and Engineering Discovery Environment (XSEDE), which is supported by the National Science Foundation grant number ACI-1548562.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data will be available upon request.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Salvati, P.; Petrucci, O.; Rossi, M.; Bianchi, C.; Pasqua, A.A.; Guzzetti, F. Gender, age and circumstances analysis of flood and landslide fatalities in Italy. *Sci. Total Environ.* **2018**, *610*, 867–879. [[CrossRef](#)] [[PubMed](#)]
2. Schuster, R.L.; Highland, L. *Socioeconomic and Environmental Impacts of Landslides in the Western Hemisphere*; United States Geological Survey: Reston, VA, USA, 2001.
3. Azarafza, M.; Ghazifard, A.; Akgün, H.; Asghari-Kaljahi, E. Landslide susceptibility assessment of South Pars Special Zone, southwest Iran. *Environ. Earth Sci.* **2018**, *77*, 1–29. [[CrossRef](#)]
4. Ahmad, H.; Ningsheng, C.; Rahman, M.; Islam, M.M.; Pourghasemi, H.R.; Hussain, S.F.; Habumugisha, J.M.; Liu, E.; Zheng, H.; Ni, H. Geohazards Susceptibility Assessment along the Upper Indus Basin Using Four Machine Learning and Statistical Models. *ISPRS Int. J. Geo-Inf.* **2021**, *10*, 315. [[CrossRef](#)]
5. Chen, W.; Chen, X.; Peng, J.; Panahi, M.; Lee, S. Landslide susceptibility modeling based on ANFIS with teaching-learning-based optimization and Satin bowerbird optimizer. *Geosci. Front.* **2021**, *12*, 93–107. [[CrossRef](#)]
6. Nanekaran, Y.A.; Mao, Y.; Azarafza, M.; Kockar, M.K.; Zhu, H.-H. Fuzzy-based multiple decision method for landslide susceptibility and hazard assessment: A case study of Tabriz, Iran. *Geomech. Eng.* **2021**, *24*, 407–418.
7. Guzzetti, F.; Reichenbach, P.; Ardizzone, F.; Cardinali, M.; Galli, M. Estimating the quality of landslide susceptibility models. *Geomorphology* **2006**, *81*, 166–184. [[CrossRef](#)]
8. Conte, E.; Pugliese, L.; Troncone, A. Post-failure analysis of the Maierato landslide using the material point method. *Eng. Geol.* **2020**, *277*, 105788. [[CrossRef](#)]

9. Kargar, P.; Osouli, A.; Stark, T.D. 3D analysis of 2014 Oso landslide. *Eng. Geol.* **2021**, *287*, 106100. [[CrossRef](#)]
10. Yang, H.; Yang, T.; Zhang, S.; Zhao, F.; Hu, K.; Jiang, Y. Rainfall-induced landslides and debris flows in Mengdong Town, Yunnan Province, China. *Landslides* **2020**, *17*, 931–941. [[CrossRef](#)]
11. Uyeturk, C.E.; Huvaj, N.; Bayraktaroglu, H.; Huseyinpasaoglu, M. Geotechnical characteristics of residual soils in rainfall-triggered landslides in Rize, Turkey. *Eng. Geol.* **2020**, *264*, 105318. [[CrossRef](#)]
12. Kirschbaum, D.; Kapnick, S.; Stanley, T.; Pascale, S. Changes in extreme precipitation and landslides over High Mountain Asia. *Geophys. Res. Lett.* **2020**, *47*, e2019GL085347. [[CrossRef](#)]
13. Jakob, M.; Lambert, S. Climate change effects on landslides along the southwest coast of British Columbia. *Geomorphology* **2009**, *107*, 275–284. [[CrossRef](#)]
14. Kristo, C.; Rahardjo, H.; Satyanaga, A. Effect of variations in rainfall intensity on slope stability in Singapore. *Int. Soil Water Conserv. Res.* **2017**, *5*, 258–264. [[CrossRef](#)]
15. Pham, B.T.; Jaafari, A.; Nguyen-Thoi, T.; Van Phong, T.; Nguyen, H.D.; Satyam, N.; Masroor, M.; Rehman, S.; Sajjad, H.; Sahana, M. Ensemble machine learning models based on Reduced Error Pruning Tree for prediction of rainfall-induced landslides. *Int. J. Digit. Earth* **2021**, *14*, 575–596. [[CrossRef](#)]
16. Sun, D.-M.; Zang, Y.-G.; Semprich, S. Effects of airflow induced by rainfall infiltration on unsaturated soil slope stability. *Transp. Porous Media* **2015**, *107*, 821–841. [[CrossRef](#)]
17. Cho, S.E. Stability analysis of unsaturated soil slopes considering water-air flow caused by rainfall infiltration. *Eng. Geol.* **2016**, *211*, 184–197. [[CrossRef](#)]
18. Alsubal, S.; Sapari, N.; Harahap, S. The Rise of groundwater due to rainfall and the control of landslide by zero-energy groundwater withdrawal system. *Int. J. Eng. Technol.* **2018**, *7*, 921–926. [[CrossRef](#)]
19. Su, Z.; Wang, G.; Wang, Y.; Luo, X.; Zhang, H. Numerical simulation of dynamic catastrophe of slope instability in three Gorges reservoir area based on FEM and SPH method. *Nat. Hazards* **2021**, 1–16. [[CrossRef](#)]
20. Ng, C.W.W.; Shi, Q. A numerical investigation of the stability of unsaturated soil slopes subjected to transient seepage. *Comput. Geotech.* **1998**, *22*, 1–28. [[CrossRef](#)]
21. Wang, J.-g.; Liang, B. Affection of rainfall factor to seepage and stability of loess slope. *J. Water Resour. Water Eng.* **2010**, *21*, 42–45.
22. Merzdorf, J. *Climate Change Could Trigger More Landslides in High Mountain Asia*; NASA's Goddard Space Flight Center: Greenbelt, MD, USA, 2020.
23. Nicholson, P.G. *Soil Improvement and Ground Modification Methods*; Butterworth-Heinemann: Oxford, UK, 2014.
24. Turner, A.K.; Schuster, R.L. *Landslides: Investigation and Mitigation*; Special Report 247; Transportation Research Board national academy Press: Washington, DC, USA, 1996.
25. Dai, F.; Lee, C.; Ngai, Y.Y. Landslide risk assessment and management: An overview. *Eng. Geol.* **2002**, *64*, 65–87. [[CrossRef](#)]
26. Urciuoli, G.; Pirone, M. Subsurface drainage for slope stabilization. In *Landslide Science and Practice*; Springer: Berlin/Heidelberg, Germany, 2013; pp. 577–585.
27. Holtz, R.D.; Schuster, R.L. *Landslides: Investigation and Mitigation*. *Transp. Res. Board Spec. Rep.* **1996**, *247*, 439–473.
28. Cashman, P.M.; Preene, M. *Groundwater Lowering in Construction: A Practical Guide*; CRC Press: Boca Raton, FL, USA, 2001.
29. Olcese, A.; Vescovo, C.; Boni, S.; Giusti, G. Stabilisation of a landslide with submerged motor-driven pumps. In *Proceedings of Slope stability engineering developments and applications*. In *Proceedings of the International Conference on Slope Stability*, Isle of Wight, UK, 15–18 April 1991; pp. 321–326.
30. Forrester, K. *Subsurface Drainage for Slope Stabilization*; ASCE Press: Reston, VA, USA, 2001.
31. Mitchell, R.J.; Madsen, J.D.; Crawford, T.W. Hydraulic stabilization of earth structures. *Can. Geotech. J.* **1984**, *21*, 116–124. [[CrossRef](#)]
32. Woodward, J. *An Introduction to Geotechnical Processes*; CRC Press: Boca Raton, FL, USA, 2005.
33. Biniyaz, A.; Azmoon, B.; Liu, Z. Coupled transient saturated–unsaturated seepage and limit equilibrium analysis for slopes: Influence of rapid water level changes. *Acta Geotech.* **2021**, 1–18. [[CrossRef](#)]
34. Wartalska, K.; Kaźmierczak, B.; Nowakowska, M.; Kotowski, A. Analysis of hyetographs for drainage system modeling. *Water* **2020**, *12*, 149. [[CrossRef](#)]
35. Chen, Y.-L.; Liu, G.-Y.; Li, N.; Du, X.; Wang, S.-R.; Azzam, R. Stability evaluation of slope subjected to seismic effect combined with consequent rainfall. *Eng. Geol.* **2020**, *266*, 105461. [[CrossRef](#)]
36. Liu, Z.L. Multiphysics in Porous Materials. In *Multiphysics in Porous Materials*; Springer: Berlin/Heidelberg, Germany, 2018; pp. 29–34.
37. van Genuchten, M.T. A closed-form equation for predicting the hydraulic conductivity of unsaturated soils 1. *Soil Sci. Soc. Am. J.* **1980**, *44*, 892–898. [[CrossRef](#)]
38. Sethi, R.; Di Molletta, A. *Groundwater Engineering: A Technical Approach to Hydrogeology, Contaminant Transport and Groundwater Remediation*; Springer International Publishing: Cham, Switzerland, 2019.
39. Xu, X.; He, H.; Zhao, D.; Sun, S.; Busoniu, L.; Yang, S.X. *Machine Learning with Applications to Autonomous Systems*; Hindawi: London, UK, 2015.
40. Wei, Z.-L.; Lü, Q.; Sun, H.-y.; Shang, Y.-Q. Estimating the rainfall threshold of a deep-seated landslide by integrating models for predicting the groundwater level and stability analysis of the slope. *Eng. Geol.* **2019**, *253*, 14–26. [[CrossRef](#)]
41. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [[CrossRef](#)]

42. Tian, Y.; Zhang, K.; Li, J.; Lin, X.; Yang, B. LSTM-based traffic flow prediction with missing data. *Neurocomputing* **2018**, *318*, 297–305. [[CrossRef](#)]
43. Duan, Y.; Lv, Y.; Wang, F.-Y. Travel time prediction with LSTM neural network. In Proceedings of the 2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC), Rio de Janeiro, Brazil, 1–4 November 2016; pp. 1053–1058.
44. Zhang, J.; Zhu, Y.; Zhang, X.; Ye, M.; Yang, J. Developing a Long Short-Term Memory (LSTM) based model for predicting water table depth in agricultural areas. *J. Hydrol.* **2018**, *561*, 918–929. [[CrossRef](#)]
45. Le, X.-H.; Ho, H.V.; Lee, G.; Jung, S. Application of long short-term memory (LSTM) neural network for flood forecasting. *Water* **2019**, *11*, 1387. [[CrossRef](#)]
46. Kaneko, R.; Nakayoshi, M.; Onomura, S. Rainfall Prediction by a Recurrent Neural Network Algorithm LSTM Learning Surface Observation Data. *AGU Fall Meet. Abstr.* **2019**, *2019*, GC43D-1354.
47. Zhang, D.; Holland, E.S.; Lindholm, G.; Ratnaweera, H. Enhancing operation of a sewage pumping station for inter catchment wastewater transfer by using deep learning and hydraulic model. *arXiv* **2018**, arXiv:1811.06367.
48. Hu, Y.; Yan, L.; Hang, T.; Feng, J. Stream-Flow Forecasting of Small Rivers Based on LSTM. *arXiv* **2020**, arXiv:2001.05681.
49. Xie, P.; Zhou, A.; Chai, B. The application of long short-term memory (LSTM) method on displacement prediction of multifactor-induced landslides. *IEEE Access* **2019**, *7*, 54305–54311. [[CrossRef](#)]
50. Yunpeng, L.; Di, H.; Junpeng, B.; Yong, Q. Multi-step ahead time series forecasting for different data patterns based on LSTM recurrent neural network. In Proceedings of the 2017 14th Web Information Systems and Applications Conference (WISA), Liuzhou, China, 11–12 November 2017; pp. 305–310.
51. Crivellari, A.; Beinart, E. LSTM-based deep learning model for predicting individual mobility traces of short-term foreign tourists. *Sustainability* **2020**, *12*, 349. [[CrossRef](#)]
52. Chung, J.; Gulcehre, C.; Cho, K.; Bengio, Y. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv* **2014**, arXiv:1412.3555.
53. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444. [[CrossRef](#)]
54. Ismail, A.A.; Wood, T.; Bravo, H.C. Improving Long-Horizon Forecasts with Expectation-Biased LSTM Networks. *arXiv* **2018**, arXiv:1804.06776.
55. Nguyen, Q.H.; Ly, H.-B.; Ho, L.S.; Al-Ansari, N.; Le, H.V.; Tran, V.Q.; Prakash, I.; Pham, B.T. Influence of data splitting on performance of machine learning models in prediction of shear strength of soil. *Math. Probl. Eng.* **2021**, *2021*. [[CrossRef](#)]
56. Bui, D.T.; Pradhan, B.; Lofman, O.; Revhaug, I.; Dick, O.B. Landslide susceptibility mapping at Hoa Binh province (Vietnam) using an adaptive neuro-fuzzy inference system and GIS. *Comput. Geosci.* **2012**, *45*, 199–211.
57. Vasu, N.N.; Lee, S.-R. A hybrid feature selection algorithm integrating an extreme learning machine for landslide susceptibility modeling of Mt. Woomyeon, South Korea. *Geomorphology* **2016**, *263*, 50–70. [[CrossRef](#)]
58. Jin, J.; Li, M.; Jin, L. Data normalization to accelerate training for linear neural net to predict tropical cyclone tracks. *Math. Probl. Eng.* **2015**, *2015*. [[CrossRef](#)]
59. Patro, S.; Sahu, K.K. Normalization: A preprocessing stage. *arXiv* **2015**, arXiv:1503.06462. [[CrossRef](#)]
60. Chai, T.; Draxler, R.R. Root mean square error (RMSE) or mean absolute error (MAE)?—Arguments against avoiding RMSE in the literature. *Geosci. Model Dev.* **2014**, *7*, 1247–1250. [[CrossRef](#)]
61. Dong, M.; Wu, H.; Hu, H.; Azzam, R.; Zhang, L.; Zheng, Z.; Gong, X. Deformation Prediction of Unstable Slopes Based on Real-Time Monitoring and DeepAR Model. *Sensors* **2021**, *21*, 14. [[CrossRef](#)]
62. Zhang, D. A coefficient of determination for generalized linear models. *Am. Stat.* **2017**, *71*, 310–316. [[CrossRef](#)]