A Smart Agriculture Solution Includes Intelligent Irrigation and Security

Tang Nguyen-Tan[1,2], Chien Dang-Ngoc[1,2], and Quan Le-Trung[1,2(✉)]

[1] Faculty of Computer Networks and Communications, University of Information Technology, Ho Chi Minh City, Vietnam
{19522181,19520424}@gm.uit.edu.vn, quanlt@uit.edu.vn
[2] Vietnam National University, Ho Chi Minh City, Vietnam

Abstract. One of the key roles of a smart agricultural system is irrigation, which is carried out automatically, optimally, and at each stage of the growth of each crop. The optimal soil moisture data for each plant at each stage of growth that have been stored in the database, along with two forecasts of the weather and the soil moisture level for the next hour, are incorporated to propose an autonomous irrigation solution in this paper. Two Transformer deep-learning models were used to train forecasts of the weather and soil moisture. The test results demonstrate that the Transformer model is able with the same accuracy of 91.41% on the weather forecast test set and 82.06% on the soil moisture forecast test set despite having 40.62% fewer training variables than the LSTM model. As an Internet of Things system, the smart agriculture system must be safeguarded against eavesdropping, attacks that spoof control commands, and machine learning models that are poisoned with false data. In this research, we have also proposed an end-to-end encryption and authentication solution using AES 256-bit, HMAC, along with a safe CRYSTALS-Kyber key exchange technique in the quantum age. The evaluation results show that the proposed solution can be deployed on IoT devices similar to Arduino, STM32, and Raspberry Pi 4.

Keywords: Smart Agricultural · Time-series Forecasting · Transformer · IoT Security

1 Introduction

Smart agriculture is the term used to refer to the application of technology to improve efficiency and productivity in agricultural activities. Smart irrigation is an Internet of Things (IoT) system comprising a sensor network, irrigation control components, and software for monitoring and remotely controlling the system. It is one of many uses of intelligent agriculture. Nowadays, irrigation systems are either manually controlled by the farmer or provide automatic irrigation through a timer; however, the farmer must analyze factors including soil moisture, weather, and the optimal environmental conditions for each stage of

© ICST Institute for Computer Sciences, Social Informatics and Telecommunications Engineering 2023
Published by Springer Nature Switzerland AG 2023. All Rights Reserved
N.-S. Vo and H.-A. Tran (Eds.): INISCOM 2023, LNICST 531, pp. 3–18, 2023.
https://doi.org/10.1007/978-3-031-47359-3_1

the plant's growth to set the right time for the timer. That inconvenience has contributed to our motivation to perform this research and use deep learning techniques to propose an appropriate autonomous irrigation system for each stage of crop growth.

Predicting soil moisture levels with machine learning algorithms is one way to enhance the performance of smart irrigation systems, making them more efficient and precise [1]. Numerous researchers from around the world have proposed the strategy of using deep learning models to optimize smart agricultural systems. The authors of a research paper proposed a method for predicting soil moisture levels in smart irrigation systems using fog computing, which involves combining multiple deep learning models such as Long Short-Term Memory Networks (LSTM), Recurrent Neural Networks (RNN), and General Deep Regression (GDR) [2]. The predicted soil moisture level will be obtained by running the K-Nearest Neighbor (KNN) algorithm on the output data of the three aforementioned deep-learning models. Reinforcement learning has been applied to automate the scheduling of irrigation for a tomato field, and the results have shown that the LSTM model is effective in reducing water usage, with savings ranging from 18% to 30% [3]. Another research employed a combination of LSTM, KNN, and Gradient Boosting-based Tree (GBT) models to forecast weather for optimizing irrigation water usage [4].

A smart agricultural system is an Internet of Things (IoT) system, which means it needs protection from network attacks. These attacks can include spoofing sensor data or control commands, as well as eavesdropping on the information. Such attacks can cause the system to malfunction, which could seriously affect the crop's growth. The article [5] proposes an authentication and key agreement protocol for IoT devices to update the secret key using the HMAC hash function. This protocol is designed to protect the system from eavesdropping and data tampering. The CRYSTALS-Kyber is a key encapsulation mechanism (KEM) that provides quantum-safe security with IND-CCA2 security. In the article [6], the first fine-grained implementation of the post-quantum CRYSTALS-Kyber KEM on a GPU is proposed, which can be utilized to offer key encapsulation and decapsulation for IoT systems. Another research found that it was possible to load and run the Kyber768 CPAPKE algorithm on the MULTOS Trust-Anchor for IoT [7], but there were performance issues due to polynomial multiplication and reduction operations. While the current performance may be adequate for machine-to-machine scenarios, further optimization studies are recommended to ensure the intended strength of the algorithm. However, it is challenging to put the aforementioned research into reality because they are purely theoretical.

In this work, we indicate a smart agriculture approach that combines deep learning and security technologies. Based on time series containing information on air temperature, air humidity, sunlight, pressure, wind speed, and time values, we trained two Transformer models to forecast weather and soil moisture in the upcoming hour. These two models were converted to TensorFlow Lite and deployed on a Raspberry Pi 4 that serves as a LoRa Gateway in our designed sys-

tem. In addition, we have proposed employing CRYSTALS-Kyber KEM for key exchange, HMAC for authentication, and AES for data encryption to increase the system's security. Our security solution will be used in the connection at the edge network as well as the connection over the internet between the system's devices, such as IoT devices (Arduino), the LoRa Gateway (Raspberry Pi), and Backend servers,...

This paper has the following structure: Sect. 1 introduces the overview of current research, motivation, and purpose of this study, and Sect. 2 will demonstrate the challenges encountered in constructing a practical smart Smart agriculture system in practice system. The IoT reference model and the network context of the system are described in detail in Sect. 3. In Sect. 4, we propose an automatic irrigation solution using two Transformer models to forecast weather and soil moisture. The end-to-end encryption and authentication solutions for the system are proposed in Sect. 5. The results of the two Transformer models' accuracy and the system's performance after the proposed security solution was applied are shown in Sect. 6. Section 7 ends this paper with a conclusion and future development directions.

2 Related Work

An overview of current smart agricultural innovations and the challenges involved in bringing them into use will be provided in this section.

2.1 Smart Agriculture System in Practice

In order to increase productivity and product quality in agricultural production, smart agricultural systems have received extensive research and practical application in developed countries. These systems are frequently constructed using local servers on a farm or centralized servers in the cloud. Wi-fi, NB-IoT, LoRa, Zigbee, Cellular, and other network technologies are frequently used for transmitting and receiving sensor data and control orders [8].

The majority of automatic irrigation systems only function according to the farmer's predetermined schedule; therefore, the best irrigation depends on the farmer's irrigation schedule. One drawback of existing smart irrigation systems is that an internet connection is required to access the server performing data review or rescheduling. This problem will be solved with the system model proposed in this study by placing the server in the internal network. When we need remote access to the server, we have set up a Zero-trust communication tunnel from the farmer's terminal software to the server, moreover, the farmers may still use their private network to access the web dashboard and manage the system even if their internet connection is down.

2.2 Security Challenge for IoT Systems

When bringing an IoT system into practice, the problem of securing the system to ensure it works properly is a huge challenge. If we only mention the system

we built in this study, the remote sensing data from the sensors can be spoofed, leading to the system making wrong decisions that seriously affect the growth of the crops. In addition, control commands can also be eavesdropped and then retransmitted in a control command spoofing attack. Typically, these issues only arise at the edge network layer, where very few security solutions are currently in use.

Because there are so many security issues with IoT systems in practice, in conjunction with the issues we highlight here, an article outlines those issues [9]. This research direction has received the attention of many researchers around the world, but the results from that research are difficult to apply in practice. The research applying the blockchain network technique has a disadvantage in that the cost of deploying and operating the system is very high, in addition, device authentication using the blockchain technique increases the latency of the system due to the call to execute smart contracts [10]. Another way is to use ECC in key exchange and authentication of edge devices, but this solution is not secure in the quantum era [11,12]. The solution proposed by us in this study will solve those problems.

3 IoT Reference Model

The IoT reference model that we use as the basis for constructing a security and deep learning solution is described in depth in this section. It is shown in Fig. 1. Sensors mounted on front-end IoT devices (Arduino or Waspmote block) help gather data on air temperature, air humidity, soil moisture, wind speed, etc. during remote sensing. In this case, the LoRa Gateway and Edge server is deployed in a 2-in-1 form on a Raspberry Pi 4 so that the collected data is transmitted to it through a LoRa connection. Then, the data will be stored in the local database as time series. During remote control, the LoRa connection is also used to forward commands to control the water pump and grow lights via the LoRa Gateway. The data stored in the local database will be aggregated, filtered with the necessary information, and periodically updated to the global database at the backend server using an internet connection.

The end user remote control the system through an application on the end device and the control command will be forwarded to the LoRa Gateway using the MQTT protocol via the MQTT Broker in the Edge server. In addition, the MQTT protocol is also used to update sensor data in real time to the end-user application. Since our system was developed via the agent application approach, each tenant will have their own Edge server. We have created a tunnel so that the tenant using our developed application can access the Edge server where the Web dashboard is installed to monitor and operate the system. We choose to deploy the system according to that model so that the Edge server can be placed in the internal network to help the system's services remain highly available when the internet connection is lost.

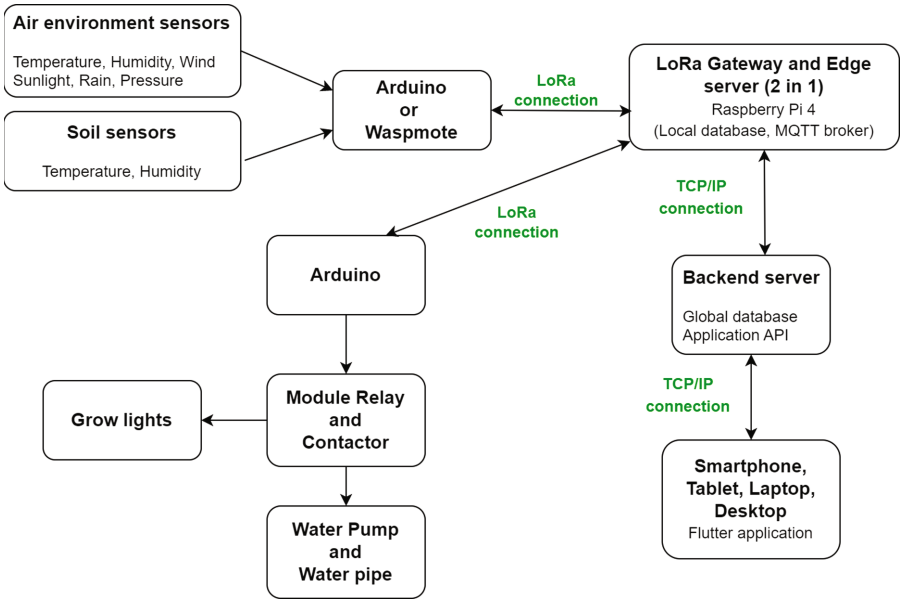


Fig. 1. IoT Reference Model.

4 Intelligent Irrigation Using Transformer Models

This section proposes two Transformer deep learning models that can be deployed on a Raspberry Pi 4 acting as a LoRa Gateway and Edge server in the IoT reference model above. The first model is used to forecast the weather, while the second model is used to forecast the soil moisture level after one hour. Both of these forecast values will then be used to schedule automatic irrigation and update the automatic irrigation schedule every five minutes. In recent years, the transformer model has attracted the attention of researchers and has been employed in much recent research to forecast time series [13–17]. We decided on using Transformer for this research rather than more traditional models like LSTM and GRU since that is a cutting-edge network architecture, even if the forecasts, in this case, are based on time series data.

4.1 Soil Moisture and Weather Forecasting

The challenge with automatic irrigation systems is to keep the soil moisture level suitable for every stage of the plant's growth. This challenge was resolved by constructing the Transformer model to forecast the next hour's soil moisture level based on five-time series containing information on soil moisture, air

temperature, air humidity, air pressure, and luminosity of sunlight collected in the previous hour. The remaining Transformer model will be used to forecast the weather for the upcoming hour because irrigation scheduling is based on both soil moisture levels and weather conditions. The weather condition will be forecasted based on five-time series that carry information on air humidity, air temperature, air pressure, wind speed, and weather conditions that were collected at previous timestamps. The forecasted value of soil moisture level and weather conditions will be used to schedule irrigation automatically.

Datasets Description

The dataset used during training and evaluation of our proposed Transformer model to forecast soil moisture is provided by SMART FASAL. This dataset of precision agriculture consists of several agricultural parameters like soil moisture, air humidity, air temperature, air pressure, and luminosity of sunlight. The interval between timestamps in the dataset ranges from one to two minutes. The dataset is conveniently available at http://smartfasal.in/ftp-dataset-portal/. In addition to forecasting soil moisture, we also construct an extra model for forecasting the weather. The dataset used to train and evaluate this Transformer model was scraped from https://www.wunderground.com with three locations in western Vietnam: Ca Mau, Rach Gia, and Can Tho City. We scraped hourly data on air temperature, air humidity, air pressure, wind speed, and weather conditions for all three locations starting on January 1, 2020, and ending on February 23, 2023, using Python and the Selenium library. The dataset is conveniently available at the Google Drive link.

Transformer Model

We constructed and trained two Transformer models based on the two datasets mentioned earlier to forecast soil moisture content and weather for the upcoming hour. The detailed structure of the two Transformer models we proposed is shown in Fig. 2. Two models are implemented based on the paper "Attention is all you need" [18] using Tensorflow with Keras modules.

Five-time series having sixty timestamps and the time signal with sixty values corresponding to those sixty timestamps serve as the input data for the soil moisture level forecasting model. The output of this model is the forecast value of the next hour's soil moisture level. In our case research, we employ sixty timestamps to forecast soil moisture levels because the interval between timestamps is between one and two minutes. The input five-time series contains data collected in the preceding hour on soil moisture, air temperature, air humidity, air pressure, and sunlight luminosity. The values of the time signal in the case of soil moisture forecasting represent the influence of the time of day on the rate of change of soil moisture.

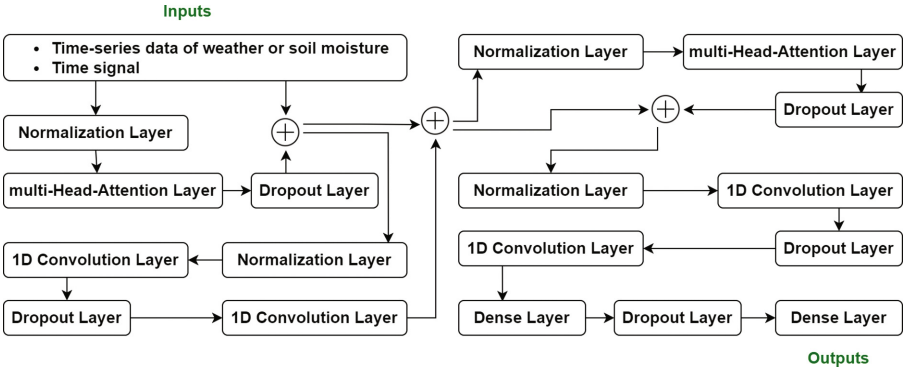


Fig. 2. Transfomer model for soil moisture and weather forecasting.

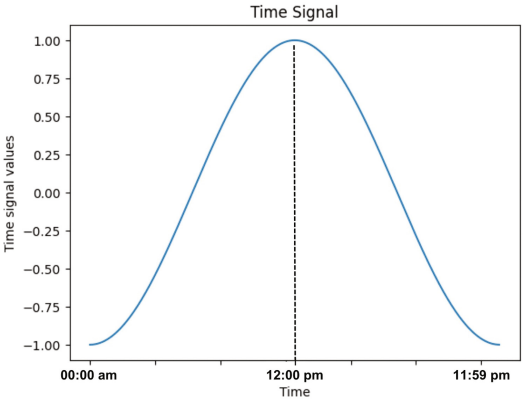


Fig. 3. The time signal.

The time signal is calculated through the formula 1 and shown in Fig. 3. In formula 1:

$$T_S = -\cos\left(\frac{60H + M + \frac{S}{60}}{1440} 2\pi\right) \tag{1}$$

- T_S is a time signal whose value ranges from -1 to 1.
- The time of day is represented by the numbers H, M, and S, respectively.

The input data of the weather forecasting model also includes the five-time series and the time signal. Five-time series containing data on weather conditions, wind speed, air pressure, air temperature, and air humidity were gathered at six timestamps earlier. The formula $T_S = \frac{N_{day}}{365}$ is used to calculate the six-time signal values matching the six timestamps in the five-time series. N_{day} is the number of days from the beginning of the year to the current time. The output of the weather forecasting model is the weather condition in the upcoming hour. Since the timestamps in the data we gathered are one hour apart, we make use of six timestamps to forecast the weather.

Following the schematic diagram of the Transformer model structure that we propose (shown in Fig. 2), the multi-Head-Attention layer in our model is multi-Self-Attention, it is used for aggregating information between timestamps while the 1D convolution layer is used for feature extraction at each timestamp. When each key's dimension is d_k and the output of the normalization layer is $\widehat{Y}$, the query, key, and value matrices are called Q, K, and V, respectively. Each

Self-Attention in our multi-Head-Attention layer is followed as formulas 2 and 3.

$$Q = \widehat{Y}W_Q \quad K = \widehat{Y}W_K \quad V = \widehat{Y}W_V \tag{2}$$

$$\textbf{Self-Attention}(Q, K, V) = \textbf{Softmax}(\frac{QK^T}{\sqrt{d_k}}V) \tag{3}$$

Two soil moisture and weather forecasting models were implemented, trained using Tensorflow, and then converted into Tensorflow Lite for deployment on Raspberry Pi 4 to carry out autonomous irrigation scheduling.

4.2 Automatic Irrigation Scheduling

This sub-section will propose an automatic irrigation scheduling technique that applies the two proposed deep learning models above. In Fig. 4, we have illustrated the specifics of the automatic scheduling method. The Raspberry Pi 4 is used to execute the diagram's orange blocks.

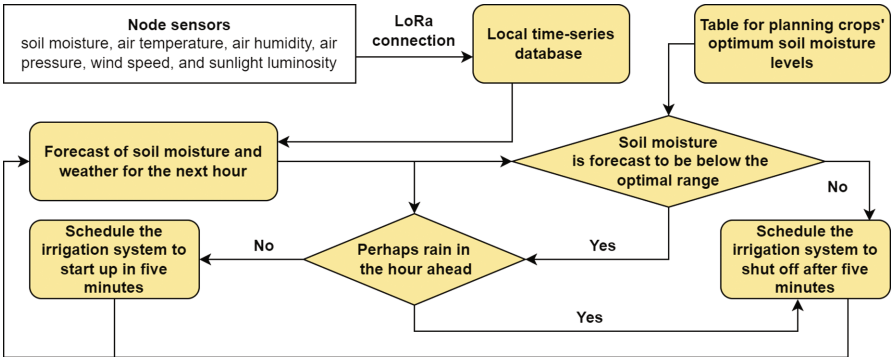


Fig. 4. Automatic irrigation scheduling diagram.

The irrigation schedule will be updated every five minutes based on the new forecast value of the weather condition and soil moisture level. When the previously scheduled time is reached, the Raspberry Pi 4 will send a corresponding control command to turn on or off the irrigation system via the LoRa connection. In addition, the end user can actively turn the irrigation system on or off via a Flutter application, shown inside the IoT reference model in Fig. 1. In case the forecasts give wrong results leading to incorrect scheduling, when the end user makes adjustments by actively turning the irrigation system on or off, the input data of the forecasts, the results of the forecasts, the scheduling results, and the control commands of the end users will be recorded and sent to the Backend server to serve the training process and improve the model.

5 End-to-End Encryption and Authentication

The intelligent irrigation method we proposed above is likely one of many other intelligent features built into contemporary smart agriculture systems that help boost efficiency and productivity in the production of agriculture. However, when implementing the system in practice, intelligent features are not enough for such an IoT system, it must be protected from spoofing attacks on sensor data, control commands, and eavesdropping data. In this section, an End-to-End encryption and authentication solution is proposed by us to apply to smart agriculture systems and can be extended to apply to other IoT systems. Our goal in our research is to deliver level 5 security in the age of quantum technology with 256 bits of safety and security. Our solution is designed to be applied synchronously on all devices of the system such as IoT devices (Arduino), Gateway (Raspberry Pi), Edge server, Cloud (Backend server), and Application.

IoT devices such as Arduino or Waspmote of Libelium, STM32 are embedded devices equipped with Micro Controller Unit (MCU) with RAM size from a few Kilobytes to several hundred Kilobytes, therefore we propose using HMAC hash with SHAKE-256 for authentication computations. There is no need to encrypt sensor data in our system because it is merely remote sensing information about the condition of the environment. However, to authenticate node sensors, we compute a digital signature and attach it to each sent data frame. Control command encryption makes little sense because they are frequently very brief and have a predetermined value, such as ON or OFF, thus we chose to add a digital signature to the control command frame as a means of authenticating its origin. The HMAC hash with SHAKE-256 is used to compute the attached digital signature in both of the sent frames stated. Authentication is necessary because our system uses LoRa waves to communicate at the edge layer. Control commands transmitted in the air environment with a radius of 3 km can be eavesdropped, and collected to perform command spoofing attacks to make the system work at the will of the attacker. The strategy of authenticating data frames exchanged between Edge Server and IoT devices is shown in Fig. 5.

Since our system was developed via the agent application approach, each tenant will have their own Edge server. Cloud not only provides data storage and backup features but also acts as an intermediary to maintain the tunnel between the edge server and the application. It is necessary to encrypt and authenticate all data transfers between the cloud, the edge server, and the application. In this case, we propose using CRYSTALS-Kyber as the encapsulation and key exchange algorithm, AES 256 bits for data encryption, and digital signature authentication using HMAC hash with SHAKE-256. We decided to use CRYSTALS-Kyber because it is a key encapsulation method (KEM) designed to be resistant to cryptanalytic attacks with future powerful quantum computers. The key exchange technique to encrypt and authenticate all data transfers between the edge server, the cloud, and the application is shown in Fig. 6.

Details of the symbols we use are shown in Table 1, the 256-byte Entropy Vector (EV) represents random values that were initially stored in the EEPROM memory of the IoT device and Edge server during system setup. The Entropy

Table 1. Symbols used in diagrams.

Symbols	Description
ID_u, ID_e, ID_c, ID_a	The identifier of IoT devices, Edge servers, Cloud, and Applications
EV_u	The 256-byte Entropy Vector
T_i	The 4-byte integer representing the time
p	The random padding
e, eu	The public entropy used to exchange a session key
K_{eu}	The shared session key between the Edge server and IoT device
SD	Data collected by sensors
CC	Control commands
tag, sig	The digital signatures
PK_c, SK_c	Cloud's public key and private key
PK_e, SK_e	Edge's public key and private key
PK_a, SK_a	Application's public key and private key
C_a	the ciphertext containing the shared session key encapsulated by the Application using Kyber
C_e	the ciphertext containing the shared session key encapsulated by the Edge server using Kyber
K_{ec}	The shared session key between the Edge server and Cloud
K_{ac}	The shared session key between the Application and Cloud
K_{ae}	The shared session key between the Application and the Edge server

Vector is random and is not the same between IoT devices so for these devices, the EV is secret and is used to compute the shared session key with the Edge server. The shared session keys are then used as input to the HMAC hash function to generate a digital signature attached to each data frame sent. T_i is a time variable used to handle data frame duplication and to generate different digital signatures for each transmission to thwart attempts to spoof sensor data and control commands.

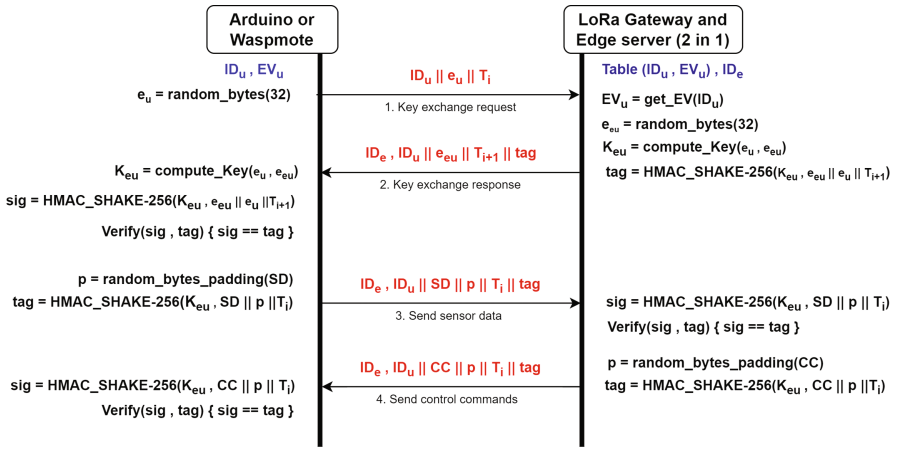


Fig. 5. Authentication strategy between the IoT device and Edge server.

In both Figs. 5 and 6 The black text denotes the node's processing tasks, the red text is the payload of the sent data frame, and the blue text is the data

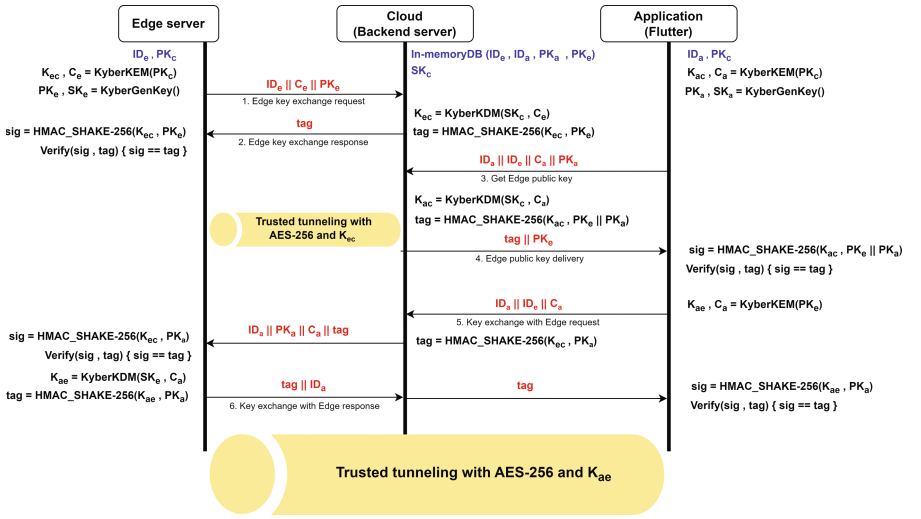


Fig. 6. Authentication and key exchange technique between Edge server, Cloud, and Application.

that has been pre-stored at the node. The following two functions are used to calculate the shared key and padding:

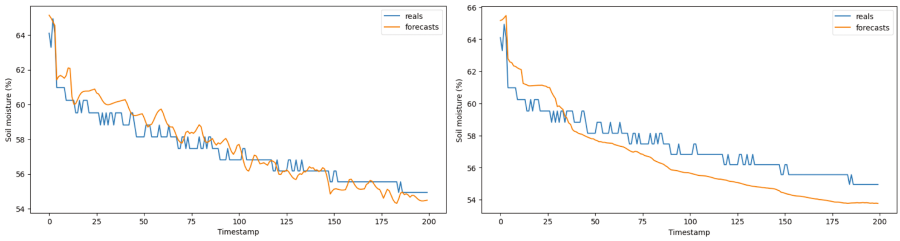
```
typedef unsigned char Byte

Byte* compute_Key(Byte* e, Byte* eu, Byte* EV) {
    Byte* pk = new Byte[64];
    for (int i = 0; i <= 32; i++) {
        pk[i] = EV[e[i]];    pk[i+32] = EV[eu[i]];
    }
    return SHAKE_256(pk);
}

Byte^ random_bytes_padding(Byte* data) {
    size_t length = length(data);
    if (length < 64) return random_bytes(64-length);
    return nullptr;
}
```

In addition to being utilized for end-to-end encryption of the data sent over the tunnel, K_{ec} and K_{ae} are employed to create digital signatures for end-to-end authentication. Every time a new session through the tunnel is formed, both of these keys are updated by using Kyber key encapsulation technique (Kyber KEM) with PK_e and PK_a.

We developed a C++ library using the CRYSTALS-Kyber official documentation [19] to implement the solution we proposed. The source code of the library we have developed can be accessed at this GitHub repository https://github.com/tangnguyendeveloper/KyberKEM_1024. In addition, we also utilized the Crypto library developed and optimized for microcontrollers like Arduino for the HMAC hash implementation with SHAKE-256, etc.

6 Experimental Results

The accuracy of the two Transformer models proposed in Sect. 4 will be evaluated along with the training outcomes in this section. Additionally, we demonstrate in this section the system's performance evaluation findings after implementing the end-to-end encryption and authentication technique we proposed in Sect. 5. Both of the early transformer models were built and trained by us with TensorFlow and then migrated to TensorFlow Lite for deployment on a Raspberry Pi 4 serving as an Edge server. The two proposed Transformer models are compared with two LSTM models that are similarly constructed to forecast soil moisture levels and weather conditions for the upcoming hour. For end-to-end encryption and authentication, we implement and evaluate the performance on devices such as Arduino uno R3, STM32F407VET6 ARM Cortex-M4, Raspberry Pi 4 Model B, Laptop (Ubuntu server 22.04 LTS), Xiaomi Redmi Note 7 (Android 10).

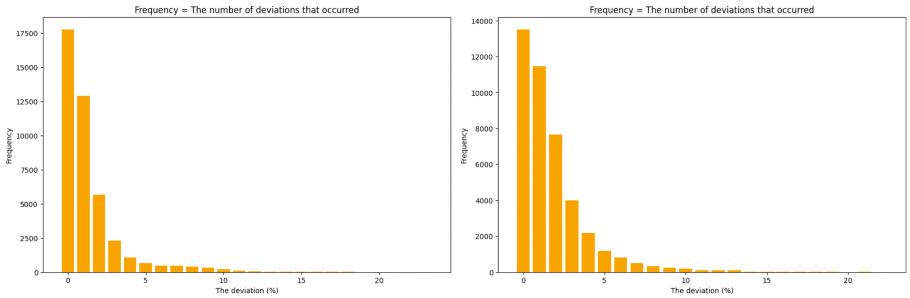


(a) Transformer (b) LSTM

Fig. 7. Compare the outcomes of the transformer model and the LSTM model forecasting soil moisture levels.

The Transformer model used to forecast soil moisture level is trained and evaluated by us on the dataset provided by SMART FASAL. Thirty percent of the samples are used by us for the accuracy evaluation, and seventy percent of the samples are used for training. Figure 7 shows the difference between the forecasted soil moisture level value with that value in the practice when getting the outcome of the Transformer model and the LSTM. Additionally, Fig. 8 depicts the distribution of the difference between the predicted value and the actual value.

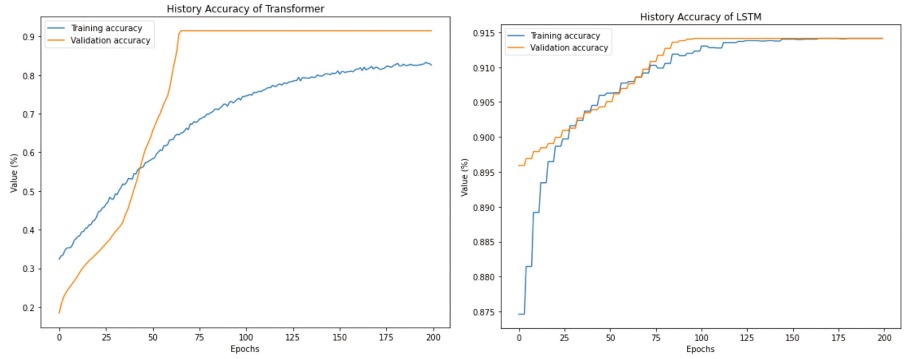
We use our scraping dataset for the training and evaluation of Transformer and LSTM models for the instance of the weather forecast. The data samples



(a) Transformer (b) LSTM

Fig. 8. Compare the frequency of the deviations for the Transformer model and LSTM model forecasts of soil moisture levels.

acquired in Rach Gia, Vietnam are utilized as the test set while the data samples collected in Can Tho and Ca Mau, Vietnam, are used in the training process. Figures 9 and 10 show the findings of a comparison of the accuracy and loss between the Transformer model and the LSTM model used for forecasting the weather conditions of the upcoming hour.

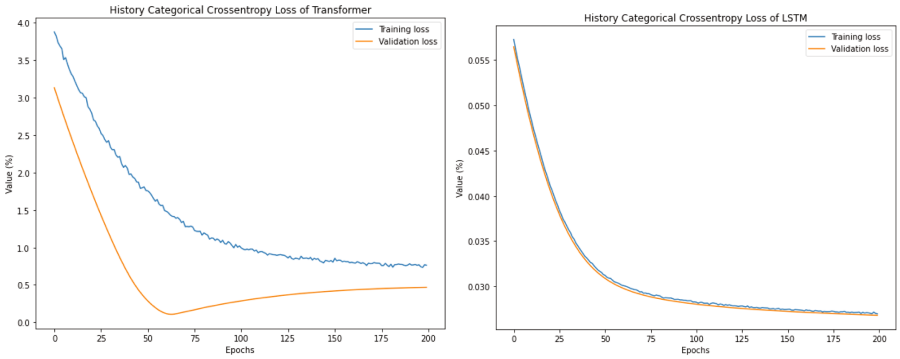


(a) Transformer (b) LSTM

Fig. 9. Compare the accuracy of the Transformer model and the LSTM model forecasting the weather conditions.

The results of evaluating the accuracy and performance of Raspberry Pi 4 when performing soil moisture forecasting and weather forecasting are shown in Table 2.

The end-to-end encryption and authentication technique proposed in this research has been implemented and evaluated by us on real devices. We used the libraries mentioned in Sect. 5 to do that, the performance benchmarks on each device are shown in Table 3. RF UART Lora SX1278 433Mhz modules were



(a) Transformer (b) LSTM

Fig. 10. Compare the Categorical Cross entropy loss of the Transformer model and the LSTM model forecasting the weather conditions.

Table 2. Summarizes the results of evaluating the accuracy and performance of soil moisture and weather forecasting models.

Model	Trainable params	epochs	Training accuracy	Testing accuracy	Performance on Raspberry Pi 4
Transformer soil moisture	30441	120	79.52%	82.06%	52 ms/sample
LSTM soil moisture	51265	120	80.17%	79.63%	108 ms/sample
Transformer weather	15306	200	82.06%	91.41%	34 ms/sample
LSTM weather	29954	200	91.41%	91.41%	71 ms/sample

Table 3. Performance benchmarks of end-to-end encryption and authentication.

	Arduino uno R3	STM32	Raspberry Pi4	Laptop	Redmi Note 7
RAM	2 KB	192 KB	8 GB	10 GB	4 GB
CPU speed	16 MHz	168 MHz	1.5 GHz	3.5 GHz	1.25 GHz
HMAC SHAKE-256	67.821 ms	7.483 ms	5.36 ns	0.627 ns	12.74 ns
Kyber1024 generate key	NaN	NaN	1.6 s	482 ms	2.217 s
Kyber1024 key encapsulation	NaN	NaN	2.235 s	739 ms	2.538 s
Kyber1024 key decapsulation	NaN	NaN	2.720 s	916 ms	3.11 s

used to implement LoRa connections for data transmission and receive in this research.

7 Conclusion and Future Work

In this research, two Transformer models were proposed to forecast soil moisture and weather conditions in the next hour to automatically schedule irrigation. The test results demonstrate that the Transformer model is able with the same accuracy of 91.41% on the weather forecast test set and 82.06% on the soil moisture forecast test set despite having 40.62% fewer training variables than the LSTM model. Not only that, an end-to-end encryption and authentication technique has also been proposed to help improve the security and reliability of

the system, protecting it from sensor data spoofing, control command spoofing, and eavesdropping attacks. In addition, we have also successfully developed a C++ library to implement the encryption and authentication technique proposed in this study. In the future, we will carry out the implementation of the proposals in this research on a practical smart agricultural system, thereby evaluating the effectiveness of the proposed solutions.

Acknowledgement. This research is funded by the Faculty of Computer Networks and Communications, University of Information Technology, Vietnam National University Ho Chi Minh City, Vietnam.

References

1. Togneri, R., et al.: Soil moisture forecast for smart irrigation: the primetime for machine learning. Expert Syst. Appl. **207**, 117653 (2022)
2. Cordeiro, M.: Towards Smart Farming: Fog-enabled intelligent irrigation system using deep neural networks. Futur. Gener. Comput. Syst. **129**, 115–124 (2022)
3. Alibabaei, K., Gaspar, P.D., Assunção, E., Alirezazadeh, S., Lima, T.M.: Irrigation optimization with a deep reinforcement learning model: case study on a site in Portugal. Agric. Water Manag. **263**, 107480 (2022)
4. Vianny, D.M.M., John, A., Mohan, S.K., Sarlan, A., Ahmadian, A.: Water optimization technique for precision irrigation system using IoT and machine learning. Sustainable Energy Technol. Assess. **52**, 102307 (2022)
5. Nakkar, M., AlTawy, R., Youssef, A.: Lightweight broadcast authentication protocol for edge-based applications. IEEE Internet Things J. **7**(12), 11766–11777 (2020)
6. Lee, W.K., Hwang, S.O.: High throughput implementation of post-quantum key encapsulation and decapsulation on GPU for Internet of Things applications. IEEE Trans. Serv. Comput. **15**(6), 3275–3288 (2021)
7. Mayes, K.: Performance evaluation and optimisation for kyber on the MULTOS IoT trust-anchor. In: 2020 IEEE International Conference on Smart Internet of Things (SmartIoT), pp. 1–8. IEEE, August 2020
8. Qazi, S., Khawaja, B.A., Farooq, Q.U.: IoT-equipped and AI-enabled next generation smart agriculture: a critical review, current challenges and future trends. IEEE Access **10**, 21219–21235 (2022)
9. Vangala, A., Das, A.K., Chamola, V., Korotaev, V., Rodrigues, J.J.: Security in IoT-enabled smart agriculture: architecture, security solutions and challenges. Cluster Comput. **26**, 1–24 (2022)
10. Khashan, O.A., Khafajah, N.M.: Efficient hybrid centralized and blockchain-based authentication architecture for heterogeneous IoT systems. J. King Saud Univ.-Comput. Inf. Sci. **35**(2), 726–739 (2023)
11. Kumar, P., Bhushan, S., Kumar, M., Alazab, M.: Secure key management and mutual authentication protocol for wireless sensor network by linking edge devices using hybrid approach. Wirel. Pers. Commun., 1–23 (2023)
12. Nakkar, M., AlTawy, R., Youssef, A.: GASE: a lightweight group authentication scheme with key agreement for edge computing applications. IEEE Internet Things J. **10**(1), 840–854 (2022)
13. Zeng, A., Chen, M., Zhang, L., Xu, Q.: Are transformers effective for time series forecasting? arXiv preprint arXiv:2205.13504 (2022)

14. Woo, G., Liu, C., Sahoo, D., Kumar, A., Hoi, S.: ETSformer: exponential smoothing transformers for time-series forecasting. arXiv preprint arXiv:2202.01381 (2022)
15. Wen, Q., et al.: Transformers in time series: a survey. arXiv preprint arXiv:2202.07125 (2022)
16. Zhou, H., et al.: Informer: beyond efficient transformer for long sequence time-series forecasting. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, no. 12, pp. 11106–11115, May 2021
17. Wu, N., Green, B., Ben, X., O'Banion, S.: Deep transformer models for time series forecasting: The influenza prevalence case. arXiv preprint arXiv:2001.08317 (2020)
18. Vaswani, A., et al.: Attention is all you need. In: Advances in Neural Information Processing Systems, vol. 30 (2017)
19. Avanzi, R., et al.: CRYSTALS-Kyber algorithm specifications and supporting documentation. NIST PQC Round **2**(4), 1–43 (2019)