Secure distribution of neural networks in wireless sensor networks

Joo Seung Yoon, Hee Moon Kim, Gil Ju Lee, Il Hee Kim, Yongsu Park[1]
The College of Information and Communications, Hanyang University,
17 Haengdang-dong, Seongdong-gu, Seoul 133-791, Korea
E-mail:{jsyoon, hmkim, gjlee, ihkim, yongsu}@hanyang.ac.kr

Abstract

In this paper we consider the case where artificial neural network processing is securely performed over the wireless sensor network. To do this, we point out major security threats and countermeasures against them. Then, we revised Holenderski et al.'s decomposition model to support secure computing. Moreover, we refine the original model to deal with some boundary cases. The revised model shows that like the original model the horizontal decomposition is better than the vertical decomposition and that the number of the allocated lower neurons in each layer to each sensor node should be large for optimization.

1. Introduction

While wireless devices are omnipresent in our lives, wireless sensor networks (WSN) are increasingly popular to deal with many monitoring problems, such as real-time traffic monitoring, wildlife tracking, bridge safety monitoring, etc. Sensor networks consist of a large amount of sensor nodes that are typically resource-constrained devices with wireless communication and data processing capabilities. These nodes collaborate with each other to build communication networks and to accomplish their task.

In this paper we focus on distributing supervised learning in the artificial neural network over the wireless sensor network. In the sensor network, there are severe resource constraints, especially communication and memory, which is due to low computation power and short battery capacity. Hence, instead of transmitting all the sensing results to the sink node and then processing all the data in the sink node, distributing supervised learning can be considered as more favorable since communication cost can be reduced and computation cost can be dissipated over the sensor network.

To the best of our knowledge, there are numerous distributed learning methods. However, most of them assume the ordinary computing environment, not the specialized ones, one of which is the wireless sensor network. Recently, M. Holdenderski et al. presented a novel model [4] to decompose the sensor nodes to distribute the learning work in the wireless sensor networks. They proposed two decomposition methods, the horizontal decomposition and the vertical one, and they compared the computation cost and the communication cost, which results that horizontal decomposition is better than the vertical one.

However, M. Holdenderski et al.'s work does not consider security aspects. Since sensor nodes can be deployed in hostile environments, security concern is very

[1] Corresponding author.

important in WSN. Many security breaches or attacking methods have been found, such as routing attacks, communication eavesdropping/modification, impersonation, DoS (Denial-of-Service) attacks, etc. Moreover, resource constraints and wireless environments add difficulties for securing sensor networks.

In this paper we consider the case where artificial neural network processing is securly performed over the wireless sensor network. To do this, we point out major security threats and countermeasures against them. Then, we revise Holenderski et al.'s decomposition model to support secure computing. Moreover, we refine the original model to deal with some boundary cases.

Our scheme can be viewed as an approach to increase trustworthiness as well as to enhance security of the systems to be protected. The rest of this paper is organized as follows. In Section 2 we describe M. Holdenderski et al.'s work. Section 3 deals with security aspects in distributed learning in the sensor network. In Section 4 we refine the M. Holdenderski et al.'s work to ensure security and to deal with some boundary conditions. Finally we offer some conclusions in Section 5.

2. Summary of M. Holdenderski et al.'s work [4]

Assume that a neural network consists of L fully connected layers where there are N neurons in each layer. The problem is to distribute this network evenly over P sensor nodes by assigning to each node a partition of weights set and the corresponding neurons.

The cost function to minimize is the maximum communication cost and the memory overhead per sensor node, as follows:

$$Cost = max_{p \in P}(Cost_{comm}(p) + k \cdot Cost_{mem}(p)) \,. \qquad (1)$$

A neural network of L fully connected layers with N neurons in each layer can be divided into as follows. Each sensor node p corresponds to l layers ($l \leq L$) where in each layer p corresponds u upper neurons, d lower neurons and all the related weights among u and d neurons. Fig. 1 shows an example of the neural network with $L = 2$ and $N = 3$ and Fig. 2 shows an example decomposition for node p with $l = 1$, $u = 2$, and $d = 3$. Similarly, Fig. 3 shows another example decomposition for node p with $l = 2, u = 1, d = 3$.

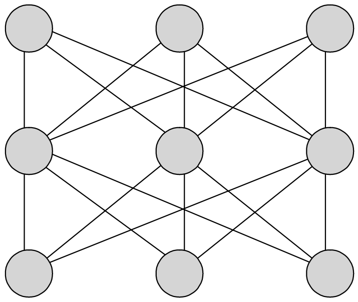

Figure 1. An example for the neural network

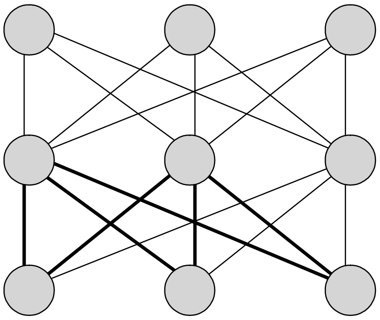

Figure 2. An example for horizontal decomposition

If the decomposition is performed with the parameter $l = 1$, we call it *horizantal* decomposition whereas if the decomposition has $l > 1$ and $u = 1$, we call it *vertical* decomposition. Fig. 2 is an example for horizontal decomposition and Fig. 3 is the example for vertical decomposition.

Since there are N^2 weights in each layer, totally LN^2 weights exist in the network. If we evenly distribute them to P sensor nodes, then the maximum number of weights assigned to each node is $E_{max} = \left\lceil \frac{LN^2}{P} \right\rceil$.

If each node p has l layers, u upper neurons, and d lower neurons, $ldu \le E_{max}$, which means that given P, N, L, d, and l, u is calculated as follows: $u \le \left\lceil \frac{E_{max}}{dl} \right\rceil = \left\lceil \frac{LN^2}{Pdl} \right\rceil$.

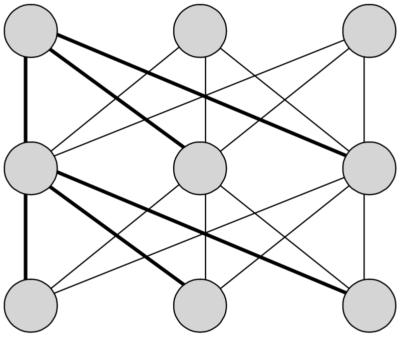

Figure 3. An example for vertical decomposition

First, consider the communication cost for each node p. The authors claim that for each layer in p there are d lower neurons exist and each of them has N inputs where for N inputs $\lceil N/d - 1 \rceil$ other sensor nodes are involved. Hence, the authors claim that each p should receive $ld\lceil N/d - 1 \rceil$ messages.

If $l > 1$ some of the partial summation between lower neurons and upper neurons can be calculated by the node p itself, which saves receiving $(l-1) \cdot min(u,d)$ messages. Consequently, the receiving communication cost for p is $R_x = ld\lceil N/d - 1 \rceil - (l-1) \cdot min(u,d)$.

As for transmission of data, since the sensor node p can broadcast the message to all the nodes, the transmission cost for p is as follows: $T_x = lu$. The total communication cost for p is $Cost_{comm} = R_x + T_x$.

The memory overhead $Cost_{mem}$ consists of the number of weights and neurons, as follows: $Cost_{mem} = E_{max} + l \cdot max(d,u) + min(d,u)$.

The authors drew some results by using $Cost_{comm}$ and $Cost_{mem}$ where as for communication cost, horizontal decomposition is always better than vertical decomposition due to reuse of data present at the sensor node. As for memory overhead, it is optimal when the upper and lower layers contain the same number of neurons.

3. Security aspects in distributed learning of ANN in the sensor network

In this section we briefly explain important security aspects in distributed learning of ANN in the sensor network, as follows.

- **Confidentiality:** In the distributed learning, computed data can be confidential. In this case, all the communication can be protected from eavesdropping. Conventional way to achive secure communication channel is to use pairwise key establishment [7, 2, 1, 3, 10, 8, 6, 5, 9]. In this method, initially a key pool which contains a large amount of symmetric keys is generated. Then, each sensor node picks a set of keys from the key pool. After all the nodes are deployed, every two neighboring nodes try to find a common key with each other pair-wisely. By using the common key, they can establish a secure channel. If a sensor node wants to communicate with non-authenticated nodes, neighboring nodes that were already authenticated act as intermediates for authentication and key exchange.
- **Data integrity:** Attackers can modify the communication, which causes incorrect calculated result. If we use pairwise key establishment, all the node share the symmetric key and by calculating message authentication codes, communication data integrity can be provided.
- **Node capturing attack:** If the attacker can capture the nodes, he can eaves some communications, which cannot be protected. However, we can protect the attack such that he tries to calculate some incorrect intermediate calculation by using multiple calculations and voting as follows. As for all the intermediate calculation, 3 nodes should perform the calculation. Only if at least 2 of them produce the same result, we can accept it.
- **Routing attack:** The adversary can attack on the routing protocol where some communication can be redirected to unintended ways or some message can be lost. If we use the pairwise key establishment and secure routing protocols, we can prevent this attack.

3.1. Classification of adversaries

In this subsection we classify adversaries as follows: random adversary, passive intelligent adversary, and active adversary.

- **Random adversary:** Random adversary is a naive attacker, who randomly select some nodes and extract key (generation) information from them. Then, he tries to eavesdrop the communication between other unattacked nodes.
- **Passive intelligent adversary:** We assume that he already has knowledge on the algorithms or protocols. Using this knowledge, he can selectively capture the nodes and then he tries to eavesdrop the communication between other unattacked nodes. However,

we assume that he does not interested in forging nodes, modify the communication messages, etc.

- **Active adversary:** In addition to the capability of passive intelligent adversary, he can use any method to interrupt/corrupt WSN, e.g., forging some nodes, jamming the communications, trying DoS (Denial-of-Service) attack, modifying the communication message, etc.

4. Refinement of the M. Holdenderski et al.'s model

In this section we describe some shortcomings in the original model and refine it. Then, we enhance the refined version to support security facilities.

4.1. Refined version of M. Holdenderski et al.'s model

Recall that in the original model, $R_x = ld\lceil N/d - 1 \rceil - (l-1)\cdot min(u,d)$. The authors claim that for each layer in p there are d lower neurons exist and each of them has N inputs where for N inputs $\lceil N/d - 1 \rceil$ other sensor nodes are involved. Hence, the authors claim that each p should receive $ld\lceil N/d - 1 \rceil$ messages. However, this may not be correct. E.g., when $N = 10, L = 2, l = 1, d = 10$, then u =2 and each node p corresponds some neurons of 1 layer having $u = 2$ upper neurons and $d = 10$ lower neurons. Because $L = 2$ layers in the network and p corresponds only 1 layer, it is evident that some communications between nodes are required. However, The equations shows that in this case $R_x = 0$, which means that there is no receiving cost, which is wrong.

Moreover, we should consider some boundary condition where if the node p has the lower neurons of the last layer, receiving cost can be reduced since there is no receiving message for those neurons. Similarly, if p has the upper neurons of the first layer, there is no transmission communications for those neurons.

Refined equations are as follows. For each layer in p there are d lower neurons and for all of them N messages from the N upper neurons of the lower layer are received. If $l > 1$ some of the partial summation between lower neurons and upper neurons can be calculated by the node p itself, hence, the receiving communication cost for p is $R_x = (l-1)(N-u) + N$ if p is not corresponding to the last layer. If p corresponds the last layer, $R_x = (l-1)(N-u)$ for $l \geq 1$.

As for transmission of data, since the sensor node p can broadcast the message to all the nodes, the transmission cost for p is as follows: $T_x = lu$ if p does not have the first layer. If p has the first layer, $T_x = (l-1)u$. The total communication cost for p is $Cost_{comm} = R_x + T_x$.

4.2. Security enhancement

As mentioned in Section 3, we can use pairwise key establishment for communication confidentiality and integrity. If we use the secure routing protocol, we can defend against the routing attack. To defend against node capturing and incorrect calculation attack, we should apply the voting protocol.

To do this, we assume that for each weight in the original network, we build 3 identical weights for the new network. the layers and neurons are identical to those of the original network. As for each decomposition, for single node in the original/revised protocol, we allocate 3 different nodes for voting. Then, results are compared each other for confirmation.

The receiving communication cost for secure version is $R_x = 3R_x + R_{voting}$. The transmitting cost for secure version is $T_x = T_x + T_{voting}$ since we use broadcasting for transmission.

The memory overhead $Cost_{mem}$ consists of the number of weights and neurons, as follows: $Cost_{mem} = E_{max} + 3(l \cdot max(d,u) + min(d,u))$.

5. Acknowledgment

This work was supported by the second stage of the Brain Korea 21 Project. Also, this work was supported by the Korea Research Foundation Grant funded by the Korean Government (MOEHRD) (KRF-2006-331-D00556). Also, this work was supported by the Korea Research Foundation Grant funded by the Korean Government (MOEHRD) (KRF-2005-042-D00294).

6. Conclusion

In this paper we consider the case where artificial neural network processing is securly performed over the wireless sensor network. To do this, we point out major security threats and countermeasures against them. Then, we revise Holenderski et al.'s decomposition model to support secure computing. Moreover, we refine the original model to deal with some boundary cases. The revised model shows that like the original model the horizontal decomposition is better than the vertical decomposition and that the number of the allocated lower neurons in each layer to each sensor node should be large for optimization.

7. Reference

[1] W. Du, J. Deng, Y. S. Han, and P. K. Varshney. A Key Predistribution Scheme for Sensor Networks Using Deployment Knowledge. *IEEE Trans. on Dependable and Secure Computing*, 3(1), 2006.

[2] W. Du, J. Deng, Y.S.Han, and P.K. Varshney. A Pairwise Key Predistribution Scheme for Wireless Sensor Networks. In *Proceedings of the 1st ACM workshop on Security of ad hoc and sensor networks*, 2003.

[3] L. Eschenauer and V.D. Gligor. A Key-Management Scheme for Distributed Sensor Networks. In *ACM CCS 2002*, pages 41{47, 2002.

[4] Mike Holenderski, Johan Lukkien, and Tham Chen Khong. Trade-o®s in the Distribution of Neural Networks in a Wireless Sensor Networks. In *First International Workshop on Data Mining in Sensor Networks*, 2005.

[5] D. Huang, M. Mehta, D. Medhi, and L. Harn. Location-aware Key Management Scheme for Wireless Sensor Networks. In *IPSN 2005*, 2005.

[6] T. Ito, H. Ohta, N. Matsuda, and T. Yoneda. A Key Pre-Distribution Scheme for Secure Sensor Networks Using Probability Density Function of Node Deployment. In *Proceedings of the 3rd ACM workshop on Security of ad hoc and sensor networks*, 2005.

[7] D. Liu and P. Ning. Establishment Pairwise keys in distributed sensor networks. In *ACM CCS 2003*, 2003.

[8] D. Liu and P. Ning. Location-based Pairwise key establishments for static sensor networks. In *Proceedings of the 1st ACM workshop on Security of ad hoc and sensor networks*, 2003.

[9] Abedelaziz Mohaisen and DaeHun Nyang. Hierarchical Grid Based Pairwise Key Predistribution Scheme for Wireless Sensor Networks. In *EWSN 2006, LNCS 3868*, 2006.

[10] Z. Yu and Y. Guan. A Key Pre-Distribution Scheme Using Deployment Knowledge for Wireless Sensor Networks. In *EWSN 2006, LNCS 3868*, 2006.

Authors

Yongsu Park received the B.E. degree in Computer Science from Korea Advanced Institute of Science and Technology (KAIST), South Korea, in 1996. He received the M.E. degree and the Ph.D. degree in Computer Engineering from Seoul National University in 1998 and 2003, respectively. He is currently an assistant professor in the College of Information and Communications at Hanyang University, Seoul, Korea. His main research interests include computer system security, network security, and cryptography.

Juseung Yoon received a B.S. degree in Communication Engineering from Daegu University, South Korea, in 2007. He is currently M.S. in Computer Engineering form Hanyang University, Seoul, Korea. His main research interests include computer system security, network security, and cryptography.

Heemoon Kim received the B.S. degree in Computer Engineering from Hanyang University, South Korea, in 2006. He is currently M.S. in the College of Information and Communications at Hanyang University, Seoul, Korea. His main research interests include computer system security, network security and crypto analysis.

Gilju Lee received the B.S. degree in Computer Engineering from Woosuk University, South Korea, in 2006. He is currently M.S. in the College of Information and Communications at Hanyang University, Seoul, Korea. His main research interests include cryptography file system and network security.

Ilhee Kim received the B.S. degree in Computer Science from Korea National Open University, South Korea, in 2006. He is currently M.S. in the College of Information and Communications at Hanyang University, Seoul, Korea. His main research interests include cryptography and network security.