# Machine operation prediction using CSI for Industrial IoT networks

AVGERINOS DEMOSTHENOUS, University of Twente, The Netherlands

Modern manufacturing facilities are increasingly utilizing industrial Internet of Things networks, these networks enable various machines and devices to establish connections, thereby enhancing operational efficiency. One crucial aspect of these types of networks is the ability to predict machine operations allowing them to optimize schedules and improve overall productivity.

This research paper will explore a novel idea of utilising Channel State Information in Industrial Internet of Things networks for the purpose of predicting the operational state of machines, specifically distinguishing between the set-up, active, and off states. Additionally, this research aims to explore optimal node positioning for the machine used and assess the impact of the environment on the model's accuracy.

To aid in this research, data is gathered from an industrial setting through uplink channels. The Channel State Information phase is utilized, and a Convolutional Neural Network is employed to analyze this information. The network classifies the collected data and predicts the state of the machine. To evaluate the effectiveness of the proposed approach, the approach is evaluated on average testing accuracy, precision, recall and f1-score.

The findings demonstrate that the states can be accurately classified with an accuracy close to 90%. The optimal node position for this experiment is the southeast position relative to the machine. Furthermore, the study confirms that the environment has an influence on the experimental accuracy.

Additional Key Words and Phrases: WiFi Channel State Information (CSI), Industrial IoT (IIoT), Convolutional Neural Network(CNN)

## 1 INTRODUCTION

The Fourth Industrial Revolution (4IR) represents the integration of previous revolutionary eras, encompassing physical and digital domains. This latest generation has exerted a profound impact on numerous industries worldwide, fundamentally reshaping their operational frameworks. Simultaneously, the advent of the Internet of Things (IoT) has emerged alongside the 4IR, giving rise to the Industrial IoT (IIoT), a convergence that synergizes the technological advancements of 4IR with the capabilities of IoT. The advent of the Industrial Internet of Things (IIoT) has opened up new possibilities for establishing a wireless network that connects IoT sensor devices. This network enables the seamless collection of data from these devices to collect data continuously and share it over the network for the purpose of maintenance [2]. Later this data could be processed to detect if there is any problem with the machines in the lab.

The emergence of wireless communication technologies in IIoT, allows CSI to be used in IIoT. CSI encompasses signal scatter, environmental attenuation, and distance, and also functions as a reference signal for characterizing wireless communication link attributes [6]. CSI demonstrates remarkable stability with minimal environmental impact and delivers satisfactory outcomes [7].

Even though there has been research on CSI used in IIoT[3], there is little to no literature on how CSI can be used in finding machine states(off, set-up, and on) in an industrial environment. According to the author's best knowledge, the experiment has never been attempted, hence there is a gap between the theory behind the paper and the reality of the results and how achievable they are.

Machine operation prediction in IIoT networks may benefit from CSI data to a number of benefits. First of all, it sheds light on the machine's physical surroundings by capturing the impacts of interference, obstacles, and multipath propagation. Second, since CSI data is easily accessible through the wireless communication infrastructure, it may be collected without requiring extra sensors or gear. This enables scalable and affordable implementation in already-existing IIoT systems.

This study aims to explore the utilization of CSI within an industrial context to predict the operational state of machines (off, set-up, and on) and determine the optimal placement of Nodes to maximize accuracy. The methodology employed for data collection draws inspiration from the experiment conducted by Anbalagan et al. [1]. To effectively analyze the collected data, a Convolutional Neural Network (CNN) will be utilized, inspired by the work of Begave et al. [1], who achieved high accuracy in motor speed classification using this type of deep learning algorithm.

The findings of this experiment demonstrate that the CNN model can achieve accuracies nearing 90% across two of the three Nodes in both controlled and uncontrolled environments. This allows for the differentiation of optimal Node placement and evaluation of the method's resilience in crowded environments. The primary contribution of this research is to showcase the practical application of CSI data for predicting machine states within an industrial setting.

The rest of the paper is organized as follows, Section 2 presents an overview of available literature related to CSI works. The experimental setup and the data acquisition are discussed in Section 3. The methodology employed for data analysis is elaborated upon in Section 4. Moving forward, Section 5 presents the results and discussion, providing a deeper understanding of the implications and significance of the findings. Section 6, outlines potential avenues for future research. Lastly, Section 7 presents the conclusion of the research.

## 2 RELATED WORK

To the author's best knowledge identifying machine states using CSI data is a novel idea, hence why there is no referenced literature on the exact concept.

As mentioned in the previous section, CSI can be used as a reference signal to describe the characteristics of a wireless communication link [5], and it can provide more stability and less environmental disturbance while still yielding satisfactory result rates.

*TScIT 39, July 8, 2023, Enschede, The Netherlands*

While CSI has been successfully used in conjunction with CNN to classify speeds in small motors, it has been observed to be sensitive to small variations in speeds under ideal network conditions [2]. Additionally, the application of this technique in an industrial environment remains an area for future exploration [2]. This study differs from the mentioned experiment in two key aspects: the use of a metal lathe machine in a metal laboratory setting with controlled and uncontrolled environments instead of small motors in a controlled environment. Nonetheless, the study provides valuable insights into preprocessing techniques to reduce data complexity and the use of a CNN model with the collected data.

Anbalagan et al. [1] conducted an experiment that closely aligns with the present study, investigating the effects of macro processes (e.g., operation shifts) and micro processes (e.g., machines with rotating components) on RSSI. Their experiments took place in the same metal workshop, and the machine used in this study was one of the machines they investigated. However, they focused on monitoring downlink data instead of uplink data. Their findings demonstrated that machine operation in a factory-like environment significantly influences the reliability of the IIoT network, resulting in a 16% variation in the packet reception rate of a node [1].

The main difference between their study and the present one lies in the focus on different metrics. While Anbalagan et al. [1] examined RSSI and downlink data, this paper focuses on CSI.

The results of Ghany et al. [5] demonstrated that CSI fingerprinting performs better than RSSI in terms of positioning and temporal stability, showing that it can be used to monitor macro processes as well.

Another notable application of CSI is the detection and classification of human activities. Damodaran et al. [4] showed that it is possible to detect and classify activities such as walking, sitting, standing, and running using CSI data. By combining the findings of Damodaran et al. [4], Bagave et al. [2], and Ghany et al. [5], it can be inferred that CSI can be utilized to identify micro and macro processes in an industrial environment, creating a more similar experiment to that of Anbalagan et al. [1].

## 3 EXPERIMENTAL SETUP AND DATA ACQUISITION

The experiment took place in one of the metal workshops at the University of Twente in The Netherlands. This workshop serves as a resource for university personnel and students to utilize machinery for their experiments during regular working hours on weekdays. The specific machine used for this experiment was a metal lathe with dimensions measuring 1.5m x 0.5m. To ensure safety, only individuals possessing the appropriate license were permitted to operate the equipment. Therefore, the experiment was conducted on two separate days when supervised access was granted.

Ideally, the workshop should have been empty or, at the very least, for individuals to be as far away as possible from the network setup in order to obtain the most desirable and unbiased results. On one of the designated experiment days, the workshop was indeed empty, allowing for the acquisition of the desired data under controlled conditions. However, on the other day, the workshop was crowded with people using various machines, which affected the accuracy of the experiment.

Normally metal lathe machines are commonly used for sharpening metallic workpieces. However, due to time constraints and the complex calculations involved in metal sharpening, it was determined that this task fell outside the scope of the project. Therefore, it was decided to proceed with the experiment by allowing the machine's "engine" to spin without any attached metal. This configuration was essential to facilitate the active phase of the experiment and gather relevant data.

In order to effectively capture the required data, a WiFi network was established around the machine. The network consisted of four Asrock NUC box-1220p devices as nodes, with one node dedicated to data transmission and the other three for reception. The experiments were performed at a 6GHz subcarrier frequency. For the communications, an injection mode setup was utilized, which allowed the transmitter to control the size of the packets, sampling rate, data rate, number of packets, and channel number for communication.

The nodes were positioned on one of the four edges of the machine. These nodes were installed using three stools and one wooden drawer, arranged according to the configuration depicted in Figure1.

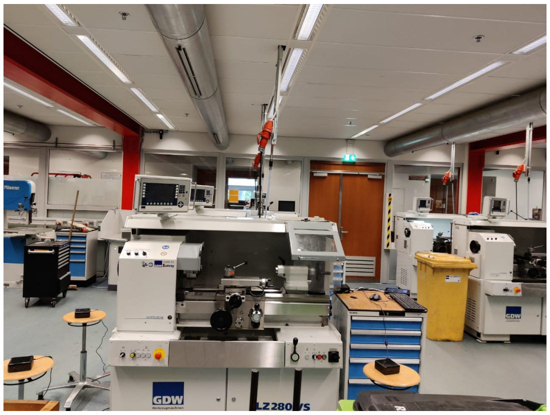


Fig. 1. Experimental setup.

The placement of the nodes was carefully chosen to ensure optimal data collection of CSI packets. The main objective of the experiment was to investigate the correlation between the CSI phase with the three distinct machine states: setup, active, and off. These states corresponded to three distinct classes, labelled 0, 1, and 2, respectively. The key difference between the active state and the other classes was the presence of vibrations at varying frequencies during proper machine operation. To accurately simulate these vibrations, different speeds were utilized since the machine did not have a metallic rod.

For the off classes, two datasets were collected each day, with varying time durations of 3-4 minutes. During this time, the machine was switched off, and efforts were made to keep people away from the network setup. For the setup class, a person stood between Nodes 2 and 3, as depicted in Figure2. The data collection time for this class was the duration required for the person to set up the machine and transition it from an off-state to a ready-to-operate state. Consequently, the dataset for the setup class was the smallest among the three classes.

Regarding the active class, two datasets were collected each day, with durations of 3-4 minutes at different speeds. Similar to the off-class, precautions were taken to prevent people from entering the network area during data collection. Subsequently, all the collected data were divided into sections of a quarter of a second for further analysis.

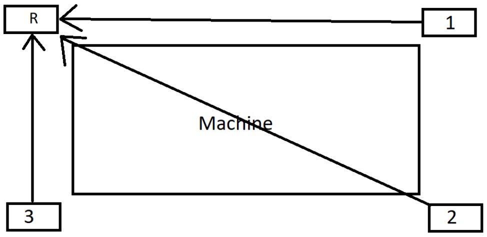


Fig. 2. Different position of notes, where R represents the Receiver

# 4 METHODOLOGY FOR DATA ANALYSIS

## 4.1 Preprocessing of the data

In the data preparation phase of the experiment, several steps were undertaken before inputting the collected CSI phase data into the classification network. The initial step involved extracting the phase information from the CSI packets, which consist of two sections: magnitude and phase. The phase component was chosen due to its higher sensitivity to small vibrations compared to magnitude.

Subsequently, the data was divided into three sections, one for each receiver, and further, split into classes: "off," "set-up," and "active". To ensure sufficient data for processing and enhance the accuracy of the CNN model, the data were segmented into sections of a quarter of a second. Furthermore, a bandpass filter was applied to the data.

Each node in the experiment comprised three antennas, two for transmitting and one for receiving. Each CSI packet contained data from 52 sub-carriers, forming a matrix of dimensions 52 sub-carriers x 2 transmitting antennas x 1 receiving antenna. Furthermore, arrays of the format (x, 52, 2, 1) were created, where 'x' denotes the amount of data collected during the experiment's duration. The sampling frequency used in the experiment was 200 Hz.

To ensure improved accuracy of the model and an adequate amount of data for processing, images were generated for the CNN by resizing the matrices. The resizing dimensions were determined as the sampling frequency (200 Hz) multiplied by the time span of a quarter of a second (0.25 sec) multiplied by the dimensions (52 x 2). Consequently, the resulting image shape was 50 x 52 x 2. This transformation aimed to provide a more comprehensive representation of the data for the CNN model.

In order to assess the reliability of the data, we meticulously examined the packet arrival rate. Our findings indicated that the observed packet arrival rate aligned with our initial expectations.

In order to reduce noise and enhance the network's accuracy, a bandpass filter was applied to all the classes (off, set-up, and active). The purpose of this filter was to eliminate human activities and environmental frequencies above the machine's vibrations. The chosen frequency range for filtering was from three to forty-two, of which three represent human activity frequencies, and forty-two corresponds to the maximum desired frequency from the machine. The value of forty-two was determined by dividing the machine's maximum spinning speed (2500) by 60 seconds and rounding up to the nearest integer, resulting in the frequency that will be filtered out.

Following the bandpass filter, a data normalization process was implemented to eliminate data redundancy and reduce processing time. The normalization technique involved scaling and adjusting the pixel values of the data. Scaling normalization ensured that each image was appropriately scaled within its own range, standardizing the values within each image independently. Pixel-level normalization further standardized the pixel values across the entire dataset, facilitating improved comparison and analysis.

## 4.2 Model Architecture

In order to prevent overfitting and monitor the training process, a k-fold cross-validation technique with five folds was utilized. Additionally, to ensure that the model was tested with unseen data, 20% of the data was randomly set aside as a separate test set to evaluate the training accuracy.

For this experiment, three models were developed, each with a different number of convolutional layers: two, three, and four. The network architecture that yielded the best results was the one with three convolutional layers. The model with four layers proved to be too deep and complex for the given problem, leading to overfitting even with various techniques applied to mitigate the issue. The two and three-layered models were comparable, but the one with three layers ultimately produced better results. Further discussion about the two and three-layered networks will be presented in the results and discussion section.

To provide a visual representation of the model, refer to Figure 4 and Figure 3 for the summary. Each convolutional layer in the model utilizes a kernel size of (3,3). This size is commonly used in CNN architectures as it is small enough to capture local patterns and edges in the input data while keeping the number of parameters manageable. Using small filter sizes in multiple layers allows the network to capture more complex patterns by stacking multiple non-linear transformations on top of each other.

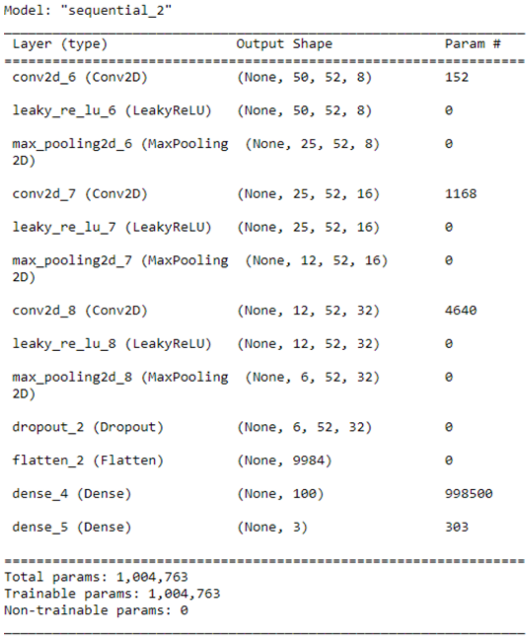
The model incorporates leaky ReLU as the activation function, with a negative gradient of 0.3. Leaky ReLU helps avoid the problem of "dead ReLU" by introducing a small negative gradient, ensuring that the network continues to learn even for negative inputs. Additionally, a MaxPooling layer is included to down-sample the features with parameters of (2,1). This downsampling operation reduces the height of the feature maps by a factor of 2 while keeping the width unchanged. This approach allows for the capture of important features along the vertical axis while maintaining resolution along the horizontal axis.

To prevent overfitting, a dropout layer with a value of 0.8 is included in the network. Dropout randomly sets a fraction of input units to 0 during training, which helps prevent the model from relying too heavily on specific features and promotes better generalization.

After the convolutional layers and dropout layer, the data are flattened. This flattening operation connects the output of the convolutional layers to dense layers, enabling the network to learn high-level features and make predictions based on the extracted information. Following flattening, there are two dense layers. The first layer is an Exponential Linear Unit (ELU) layer with one hundred neurons, which incorporates L1 and L2 regularization with values of 0.1 and 0.3, respectively. The ELU activation function introduces non-linearity to the network and aids in capturing complex patterns in the data. Although the model could have used Rectified Linear Unit (ReLU), leaky ReLU and ELU were employed to avoid the dead ReLU problem by introducing a negative gradient.

L1 regularization helps minimize overfitting by adding a penalty to the loss function. This penalty encourages some coefficients to become zero, effectively removing less important features from the model. By reducing the number of features, the model avoids memorizing noise or less significant patterns in the training data, resulting in improved generalization and reduced overfitting.

L2 regularization also mitigates overfitting and enhances model performance by introducing a penalty that discourages the model from relying too heavily on a few features. Finally, the network concludes with a SoftMax dense layer with three neurons, corresponding to the three states being classified.



```
Model: "sequential_2"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_6 (Conv2D)           (None, 50, 52, 8)         152

 leaky_re_lu_6 (LeakyReLU)   (None, 50, 52, 8)         0

 max_pooling2d_6 (MaxPooling  (None, 25, 52, 8)        0
 2D)

 conv2d_7 (Conv2D)           (None, 25, 52, 16)        1168

 leaky_re_lu_7 (LeakyReLU)   (None, 25, 52, 16)        0

 max_pooling2d_7 (MaxPooling  (None, 12, 52, 16)       0
 2D)

 conv2d_8 (Conv2D)           (None, 12, 52, 32)        4640

 leaky_re_lu_8 (LeakyReLU)   (None, 12, 52, 32)        0

 max_pooling2d_8 (MaxPooling  (None, 6, 52, 32)        0
 2D)

 dropout_2 (Dropout)         (None, 6, 52, 32)         0

 flatten_2 (Flatten)         (None, 9984)              0

 dense_4 (Dense)             (None, 100)               998500

 dense_5 (Dense)             (None, 3)                 303

=================================================================
Total params: 1,004,763
Trainable params: 1,004,763
Non-trainable params: 0
_________________________________________________________________
```

Fig. 3. Summary of the Network.

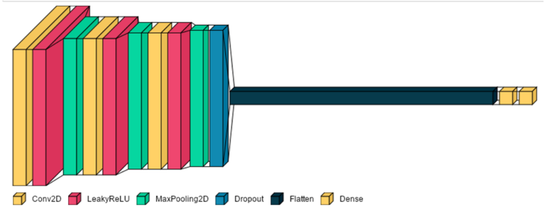


Fig. 4. Visualisation of the Network.

## 5 RESULTS AND DISCUSSION

In the upcoming section, the testing outcomes, including precision-recall and F1 scores, will be presented for the uplink data of the CSI phase collected on day one and day two. Additionally, the rationale for choosing the three-convolutional layer network over the two-layered network will be discussed.

### 5.1 Model comparison

By comparing Figures 5 and 6 with Figures 7 and 8, we observe that the average accuracy during training is slightly higher when employing a three convolutional layer network as compared to a two convolutional layer network. However, it is important to note that higher accuracy alone does not necessarily indicate the superiority of one model over the other. The next step involves evaluating whether any of the models exhibit overfitting or underfitting and then comparing precision, recall, and F1 scores.

Throughout the training process, both models demonstrate similar loss values, with the two-layered model displaying slightly lower accuracy. However, neither model shows signs of overfitting or underfitting. By comparing Table 1 and Table 2 with Table 3 and Table 4, it becomes evident that the three-layered network consistently yields better results compared to the two-layered network. Hence,

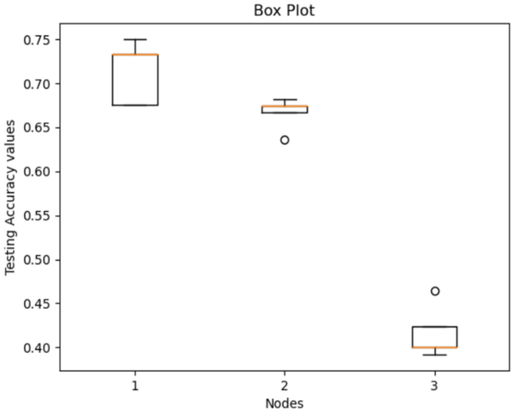


Fig. 5. Box plot of the two convolutional layer Day 1.

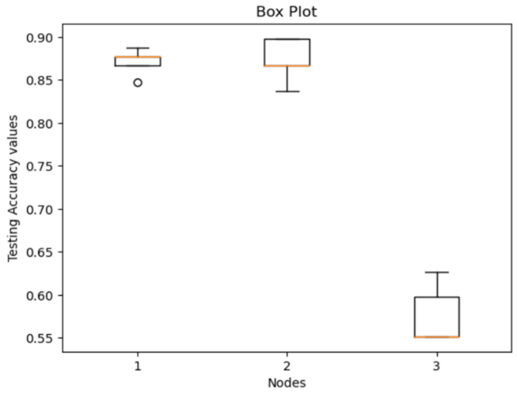


Fig. 6. Box plot of the two convolutional layer network Day 2.

the three-layered network is selected as the preferred model for this experiment.

## 5.2 Nodes position

It is worth noting that the same CNN model is used for both days, and its performance varies when confronted with unseen data during the testing phase. The results clearly indicate that the model's accuracy is influenced by the position of the node as well as the activities occurring in the laboratory.

As mentioned earlier, each node is associated with three classes: "set-up" represented by 0, "on" represented by 1, and "off" represented by 2. The positions of each node are depicted in Figure 2. The following figures present the average testing values for the data collected on both days, while the tables provide precision, recall, and F1 scores for each class of every node.

On the first day, the data collection was conducted to assess if there were any discernible differences in the results without considering the positions of the nodes. As depicted in Figure 7, Node 1 and Node 2 exhibit similar average accuracies, both close to 80%. This suggests that their positions are good candidates for collecting data compared to Node 3, which has an average accuracy of 60

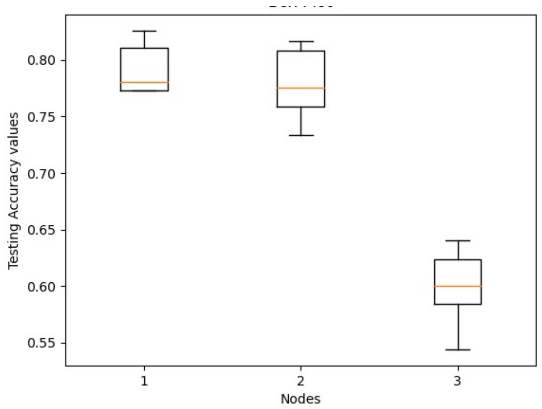


Fig. 7. Box plots of the average testing accuracy of Day 1 collection.

Moving on to the second day, as shown in Figure 8, Node 1 demonstrates significantly higher accuracy compared to any other node in the experiment, with an average close to 95%. This notable increase can be attributed to a person walking through the network and using the machine in front of the monitored machine. It appears that the signal from Node 1 was monitoring the experimental machine, the person, and possibly the machine that the person was working on. Node 2 achieves an average accuracy close to 90%, while Node 3 has an average of 75%, making it the least favourable position among the three.

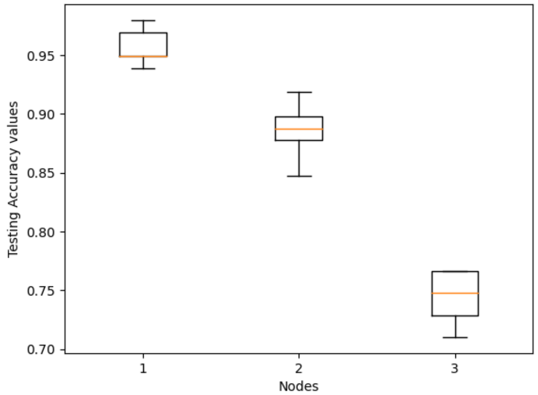


Fig. 8. Box plots of the average training accuracy of Day 2 collection.

Furthermore, by referring to Table 3 and Table 4, Node 2 exhibits scores similar to Node 1, and in some cases, it even outperforms Node 1, making Node 2 the most suitable node among the Nodes for this experiment.

Table 1. Precision-Recall-f1-score Day 1 collection(2 Conv layers).

| Node 1 | | | |
|---|---|---|---|
| | Precision | Recall | f1-score |
| 0 | 0.78 | 0.80 | 0.79 |
| 1 | 0.68 | 0.48 | 0.56 |
| 2 | 0.68 | 0.86 | 0.76 |
| | | | |
| Node 2 | | | |
| 0 | 0.69 | 0.60 | 0.64 |
| 1 | 0.47 | 0.60 | 0.53 |
| 2 | 0.79 | 0.68 | 0.73 |
| | | | |
| Node 3 | | | |
| 0 | 0.40 | 0.41 | 0.41 |
| 1 | 0.47 | 0.21 | 0.30 |
| 2 | 0.42 | 0.64 | 0.51 |

To conclude, the highest accuracies were observed when using a model with three Convolutionary layers. Referring to Figure 2, the machine's engine is located on the line between point R and point 1. It is noteworthy that Node 3 consistently exhibited the worst

Table 2. Precision-Recall-f1-score Day 2 collection(2 Conv layers).

| Node 1 | | | |
|---|---|---|---|
| | Precision | Recall | f1-score |
| 0 | 0.67 | 0.75 | 0.71 |
| 1 | 0.95 | 0.58 | 0.72 |
| 2 | 0.76 | 0.97 | 0.85 |
| | | | |
| Node 2 | | | |
| 0 | 0.89 | 0.75 | 0.81 |
| 1 | 0.79 | 0.91 | 0.85 |
| 2 | 0.97 | 0.97 | 0.97 |
| | | | |
| Node 3 | | | |
| 0 | 0.59 | 0.69 | 0.63 |
| 1 | 0.54 | 0.72 | 0.62 |
| 2 | 0.72 | 0.36 | 0.48 |

Table 3. Precision-Recall-f1-score Day 1 collection(3 Conv layers).

| Node 1 | | | |
|---|---|---|---|
| | Precision | Recall | f1-score |
| 0 | 0.83 | 80 | 0.81 |
| 1 | 0.65 | 0.84 | 0.73 |
| 2 | 0.71 | 0.80 | 0.75 |
| | | | |
| Node 2 | | | |
| 0 | 0.82 | 0.90 | 0.86 |
| 1 | 0.74 | 0.57 | 0.65 |
| 2 | 0.71 | 0.8 | 0.75 |
| | | | |
| Node 3 | | | |
| 0 | 0.70 | 0.46 | 0.56 |
| 1 | 0.45 | 0.62 | 0.52 |
| 2 | 0.70 | 0.67 | 0.68 |

performance. This outcome was expected because the path from Node 3 to the receiver has limited interaction with the machine, resulting in minimal changes in the CSI phase. On the other hand, Nodes 1 and 2 demonstrated similar accuracies due to their shared path, which involves passing through the machine engine to point R. This path introduces significant vibrations, and since the CSI phase is highly sensitive to even slight deviations, better results were obtained.

Although Node 1 exhibited higher accuracies compared to the other nodes, on a busy day, its performance was affected by the environment by approximately 15%, Node 2 experienced an average impact of only 5%. Consequently, Node 2 can be considered the optimal position for this particular scenario. However, it should be noted that in a different laboratory architecture, the same position might yield different results. Thus, it is safe to assume that no model fits all.

Table 4. Precision-Recall-f1-score Day 2 collection(3 Conv layers).

| Node 1 | | | |
|---|---|---|---|
| | Precision | Recall | f1-score |
| 0 | 0.90 | 74 | 0.81 |
| 1 | 0.75 | 0.75 | 0.75 |
| 2 | 0.71 | 0.83 | 0.77 |
| | | | |
| Node 2 | | | |
| 0 | 0.79 | 0.97 | 0.87 |
| 1 | 0.97 | 0.85 | 0.9 |
| 2 | 1 | 0.91 | 0.95 |
| | | | |
| Node 3 | | | |
| 0 | 0.73 | 0.69 | 0.71 |
| 1 | 0.8 | 0.67 | 0.73 |
| 2 | 0.73 | 0.89 | 0.8 |

### 5.3 Limitations

The limited time available, the unavailability of large computational resources, and scheduling difficulties in the metal lab imposed constraints on the experiments conducted with the data. The unavailability of large computational resources played a significant role, causing delays of several hours in data processing when testing different parameters. The time limitations further restricted the exploration of additional possibilities, such as experimenting with alternative deep learning algorithms or identifying the optimal parameters for the network. Consequently, the experiment did not reach its full potential in various aspects, which will be discussed in the future work section.

Another challenge encountered was the lack of control over the environment, leading to certain imbalances in the data. However, this situation provided an opportunity to compare and analyze how the environment can impact the data. Despite the limitations posed by the uncontrolled environment, valuable insights can still be gained from understanding the effects of such factors on the experimental outcomes.

In summary, the experiment faced limitations due to limited time, hardware constraints, and uncontrolled environmental conditions. These factors hindered the exploration of additional possibilities and optimization of the experiment. However, the challenges also provided opportunities for examining the influence of the environment on the data, contributing to valuable insights and potential avenues for further research.

## 6 FUTURE WORK

For future research or replication of this experiment, several recommendations can be considered to further improve the analysis and evaluation of the proposed approach, leading to more accurate outcomes.

Firstly, increasing the duration of data collection would provide a larger dataset, allowing for more in-depth analysis and evaluation.

This would provide a more comprehensive understanding of the proposed approach and potentially yield better results.

Additionally, in this experiment, only the uplink data was utilized, and the cyclic feature of the nodes was not explored. Future work could involve incorporating the cyclic feature into the analysis, which could provide additional information on how the environment affects the network.

Comparing this experiment to Anbalagan's [2], it could be noted that the heterogeneous effects of the macro processes could have been avoided if the cyclic feature of the network had been utilized. Furthermore, a dissimilar relationship between features mentioned in Anbalagan's work was addressed in this experiment by implementing a deep learning algorithm. For better comparison with Anbalagan's experiment, future work could involve exploring the downlink data.

Another avenue for future research is to investigate whether there are better deep-learning algorithms for analyzing CSI packets. In this experiment, only a CNN was used due to time constraints. Exploring other algorithms or fine-tuning the parameters of the CNN could potentially lead to improved results.

Furthermore, conducting a more detailed investigation into how the environment affects the network would be beneficial. For example, analyzing how the presence of a human passing through the network compared to the absence of a human or the impact of having an active machine in close proximity to the network on CSI packets. Understanding these environmental factors can provide valuable insights for optimizing network performance.

Additionally, a more complex experiment could involve predicting the next stage of the machine, such as transitioning from off to idle and then to active or predicting the worker's position in the lab. Since a user operating the machine cannot leave the lab without walking, an algorithm could be developed to detect human activity in the lab.

In summary, future research directions include increasing data collection time, exploring the cyclic feature and downlink data, investigating alternative deep learning algorithms, conducting a more detailed analysis of environmental effects, and exploring more complex prediction tasks related to machine stages or worker positions in the lab.

## 7 CONCLUSION

Based on the results, we can conclude that CSI can be used to classify different machines' states when combined with a CNN model, with accuracies close to 90% in detecting machine states and finding that the best position to place a Node is position 2. However, it is important to acknowledge some of the limitations of this experiment. The sample size of collected data per Node was limited due to time constraints. Also, having an uncontrolled environment was not ideal, but this allowed for investigating how humans can affect the network. Lastly, the positioning chosen for this experiment was based on the lab's architecture, trying to surround the machine with the network while at the same time minimising the interaction with the environment; different lab architecture might interact differently with this Node positioning.

Overall this is a novel concept. A lot of things could have been done better, and there are a lot of things to investigate. I think this experiment can spark ideas for new experiments and explore CSI's full potential when it comes to monitoring in an IIoT environment.

## ACKNOWLEDGMENTS

The author gratefully acknowledges the valuable contributions of Jeroen Klein Brinke and Sabari Nathan Anbalagan, who provided insightful ideas and exceptional supervision throughout the project. Additionally, special appreciation is extended to Norbert Spikker for his assistance in setting up the experiment.

## REFERENCES

[1] Sabari Nathan Anbalagan, Fatjon Seraj, and Paul Havinga. 2020. Towards factory schedule based adaptation for reliable networking in industrial IoT. In *2020 IEEE 29th International Symposium on Industrial Electronics (ISIE)*. IEEE, 1115–1120.

[2] Prachi Bagave, Jeroen Linssen, Wouter Teeuw, Jeroen Klein Brinke, and Nirvana Meratnia. 2019. Channel state information (CSI) analysis for predictive maintenance using convolutional neural network (CNN). In *Proceedings of the 2nd Workshop on Data Acquisition To Analysis*. 51–56.

[3] Yuanfang Chen, Gyu Myoung Lee, Lei Shu, and Noel Crespi. 2016. Industrial internet of things-based collaborative sensing intelligence: framework and research challenges. *Sensors* 16, 2 (2016), 215.

[4] Neena Damodaran, Elis Haruni, Muyassar Kokhkharova, and Jörg Schäfer. 2020. Device free human activity and fall recognition using WiFi channel state information (CSI). *CCF Transactions on Pervasive Computing and Interaction* 2 (2020), 1–17.

[5] Ahmed Abdel Ghany, Bernard Uguen, and Dominique Lemur. 2020. A robustness comparison of measured narrowband csi vs rssi for iot localization. In *2020 IEEE 92nd Vehicular Technology Conference (VTC2020-Fall)*. IEEE, 1–5.

[6] Zhu Wang, Bin Guo, Zhiwen Yu, and Xingshe Zhou. 2018. Wi-Fi CSI-based behavior recognition: From signals and actions to activities. *IEEE Communications Magazine* 56, 5 (2018), 109–115.

[7] Zhengjie Wang, Kangkang Jiang, Yushan Hou, Zehua Huang, Wenwen Dou, Chengming Zhang, and Yinjing Guo. 2019. A survey on CSI-based human behavior recognition in through-the-wall scenario. *IEEE Access* 7 (2019), 78772–78793.