Short-Term Wireless Connectivity Prediction for Connected Agricultural Vehicles

Brynjar Örn Grétarsson, Charalampos Orfanidis, Letizia Marchegiani, and Xenofon Fafoutis

***Abstract*— Robots and autonomous vehicles have been integrated in our life and utilized in a plethora of application scenarios, including intelligent transportation, industrial automation and smart agriculture. Several of the these applications might be functioning in environments where cellular network coverage is low or non-existent. In a case like this, lower bandwidth networks and vehicle-to-vehicle communication can be used to keep the application operating safely, even with less active features. In such settings, disconnection events can be avoided if deteriorating communication links are detected early so that prevention measures can be taken. In this paper we investigate how we can predict if a communication link will be terminated in the near future based on the recent trend of the signal. We propose a deep neural network framework which is executed onboard and we evaluate its performance based on simulation and real word data. The results show that we can predict the termination of a link up to 7 seconds into the future with 72.38% accuracy and 86.38% recall.**

I. Introduction

In the last decade, Internet of Things (IoT) technology has been enabling numerous industrial applications such as intelligent transportation [1], Industry 4.0 [2], and smart agriculture [3]. In parallel, several of those application domains has been enhanced by advancements in robotics technology. In agriculture, for instance, collaborating fleets of autonomous agricultural vehicles (e.g. autonomous tractors) have been proposed as means for improving the productivity and efficiency of agricultural processes [4]. The synergies and challenges of integrating robots within the IoT ecosystem is recently investigated as part of the Internet of Robotic Things (IoRT) [5], [6].

Several of the aforementioned use cases are safety-critical, especially when they involve mobile systems that have the potential to harm people. Autonomous tractors, for example, pose a potential danger to people and animals that may coexist in the field. For this reason, the existence of reliable wireless connectivity from a mobile robot to other robots in the fleet and/or a human operator at a control centre is vital safety feature and often a regulatory requirement [7].

Wireless networking solutions that are designed for safety-critical industrial applications achieve high reliability by employing techniques, including time synchronisation, scheduled communication with time slot allocation, and channel hopping, to combat interference and avoid collisions [8]. The challenge, however, is that schedule distribution and joining a network are time consuming, making the network inflexible to sudden changes. As a result, industrial wireless networking solutions struggle to yield high reliability in mobile networks where mobile nodes move out of coverage [9]. This results to packet loss when the mobile node is out of coverage and significant overhead and further packet loss during the reconnection process [10].

Autonomous vehicles, like sensor networks, are cyber-physical systems that depend on traditional (*e.g.*, [11]) and nontraditional sensors (*e.g.*, [12]) to perceive the environment and act to optimise their application objectives. In contrast to wearable sensors, where the device has no control of the mobility of the user, autonomous vehicles have the ability to adjust their mobility to maximise other objectives, *e.g.*, planning based on energy consumption estimates [13], [14]. In similar spirit, we argue that a vehicle can also monitor the state of the wireless communication link and adjust its behaviour, so that the application objectives are optimised without compromising the wireless connectivity.

To this end, we propose a framework that is able to predict if a wireless connection will break in the near future based on the recent trend of the link's signal. The idea is that the output of this framework can be used as an alarm and trigger some action, such as speed or path adaptation, that would prevent a disconnection event from occurring in the first place. The main challenges in this endeavour are that the prediction should be computed on-board, in real-time, and with the potentially limited resources of a mobile embedded system. An additional challenge is the practical difficulty and cost associated with collecting large datasets from real-world mobile systems for training the model. Indeed, capturing real-world disconnection events is especially difficult, if not impossible, in scenarios where connectivity is a regulatory requirement. To address this challenge, in this paper, we propose the use of artificial data (obtained from a simulator) augmented with real-world noise. Our results suggest that the model trained with the augmented data significantly outperforms a model trained with artificial data without the enhancements, yielding performance comparable to the model trained on real-world data.

II. Related Work

In wireless networking, various metrics that are based on the strength of the received signal have been used to estimate the quality of the link. The most common one is the widely available RSSI (Received Signal Strength Indicator), but also include the more vendor-specific LQI (Link Quality

Brynjar Örn Grétarsson, Charalampos Orfanidis, and Xenofon Fafoutis are with the Department of Applied Mathematics and Computer Science, Technical University of Denmark, Denmark.

Letizia Marchegiani is with the Department of Engineering and Architecture, University of Parma, Italy.

Corresponding author: Xenofon Fafoutis (xefa@dtu.dk).

Indicator) and the SNR (Signal-to-Noise Ratio) [15], [16], [17], [18], [19], [20]. These metrics, however, focus on estimating the current quality of a wireless link; instead, our interest is in predicting how the quality of the link will develop in the near future.

More recent works use the same metrics as input to train models that predict the future link quality. In [21], the authors use the SNR to predict the expected signal strength targeting people moving in buildings. This is done by creating a signal strength map of the building and correlating the real-time measurements to that map. The main drawback of this technique is that it expects substantial data collection on the deployment environment which is assumed to be relative constant. Similarly, in [22] the authors design models that predict whether the next packet transmission will be successful. They experiment with several prediction models, and conclude in employing logistic regression for is predictive performance and relatively low processing cost. The work relies on a static network with constant environmental conditions that cannot be assumed in mobile scenarios.

In [23] and [24], the authors propose TALENT: an link quality predictor for making routing decisions. Rather than relying on past measurements, the proposed method employs online learning using a logistic regression classifier with stochastic gradient decent to predict the link quality in 1 second in the future. The key disadvantages of the proposed scheme is that it relies on high data rates and outputs estimates that are only valid for 1 second, providing insufficient time for vehicle to react to prevent the link failure.

Lastly, in [25], the authors consider a mobile robot moving indoors and use an infrastructure of multiple static receivers to predict the RSSI profile of the mobile link. While their scenario shares a lot of similarities with the one we focus on this paper, the reliance on multiple static receiver nodes makes it inapplicable to outdoor settings where such infrastructure does not exist, such as agricultural scenarios.

III. Dataset

For the purpose of this work, we generated three datasets, namely *artificial data*, *artificial data enhanced with real measurements* and *real world data*. The data are raw RSSI values while the nodes are moving to capture how a signal strength trend can be used to predict a disconnect event. The capture rate is 1 packet per second.

The artificial data is generated by using Cooja simulator [26] and unit disk graph medium with path loss. Two nodes are included, a static one, a mobile one and the mobility pattern of the later is illustrated in Fig. 1. The static node acts like a transmitter and the mobile node as a receiver. The mobile node always starts close to the static one, to be within its coverage and during the type 1 pattern it can move via the x-axis and then the y-axis or vice versa. Path type 2 can be described by moving diagonal. The probability of selecting each path, is equal. The mobile node also picks a random speed value between $1, 2, 3$ m/s. The size of the area was 160 m^2 resulting in a maximum distance of 40 m between the two nodes. In the simulator there is no

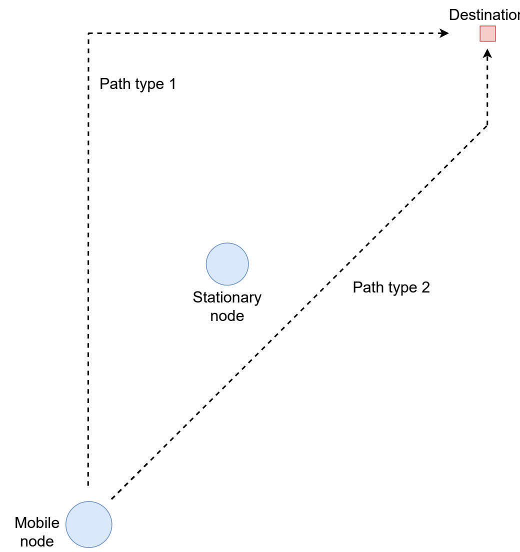

Fig. 1. The path types the mobile node can move on the evaluated scenarios.

interference, the noise floor is static, the simulated antennae are omnidirectional and it is assumed line of sight all the time. The total amount of simulation time to obtain the artificial data was approximately 950 minutes.

The real-world data obtained using two CC2650 radios [27], a static and a mobile which was following a mobility pattern similar to the one in Fig. 1. The minimum and maximum distance was 10 cm and 60 m. The mobility patterns were composed by moving at different speeds between 0.3 to 1.3 m/s. The transmission power varied between -21 to 5 dBm as we were trying to make a link to drop within the available distance. The data was collected in soft ground like dirt or grass, and not a hard one which can influence the range of the transmissions. The total amount of data is around 250 minutes of transmissions which is available online[1].

The data from the simulator was much smoother than the real-world data. Aiming for a scalable solution that does not depend on real-world data, which in some cases might be impossible to collect due to regulatory requirements, some randomness was added to the artificial data to mimic the real-world data. The artificial data enhanced by real measurements were generated by using two CC2650 radios and making some observations. More specifically, two static nodes, a transmitter and a receiver, transmitting a packet every second, were used to obtain the Mean Squared Error (MSE) of the received signal. The transmission power was 0 dB and we used multiple distance values, namely $30, 20, 10$ m, for 70 seconds each time. The results are presented in Fig. 3. Incorporating this error into the data was done by introducing a random offset to each measurement point from -2 dB to $+2$ dB, with both values included, which over a long time results in a MSE of 2 dB approximately. The effect

[1] `https://github.com/brynjarorng/Rssi-mobility-measurements`

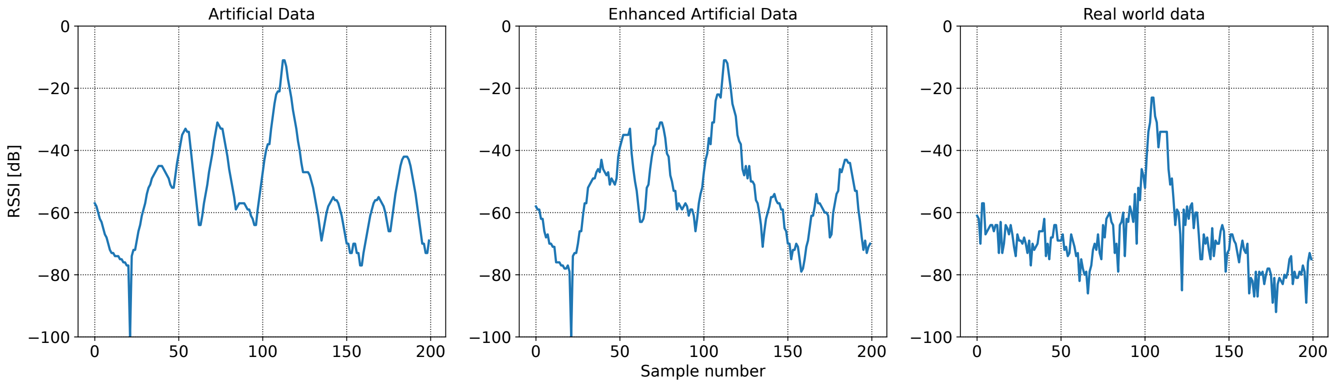

Fig. 2. Comparison between generated data and collected data.

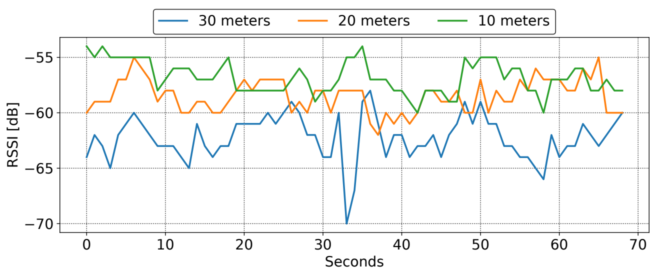

Fig. 3. The Mean Squared Error at different distances which was used to enhance the artificial data.

of this can be seen in Fig. 2.

IV. Model Design and Data pre-processing

This section discusses the selected model as all the major design choices along with the pre-processing pipeline. The overall framework is summarised in Fig. 4.

The designing process was inspired by previous research [28], [15], [29]. The final design of the current model, shown in Fig. 5, is based on a feed-forward network which consists of 10 layers where each layer includes between 5 to 300 neurons, beside the output which includes only 1. All layers are fully connected (not displayed in Fig. 5). The activation function in the input and hidden layers is a Leaky ReLU while the output activation function is a sigmoid function. There are 4 features given as input to the network and the output is either 1 or 0 denoting the likelihood of an upcoming disconnect event or not.

One of the challenges was to select an RSSI value which indicates that a signal is fading out and the Packet Reception Rate (PRR) starts to drop and use it for labelling the data. We consulted the literature [16], [20], [15], [30] and selected the value of -87 dB. The results from the literature are based on the CC2420 radio which has similar sensitivity (-95 dBm) to CC2650 radio (-97 dBm) which is the one we used. Note that in approaches with a different radio this value should be reconsidered as well as other parameters such as environmental conditions. The neural network is regularized with the dropout method to reduce overfitting. The dropout rate was set to 50% by deploying dropout layers between the fully connected dense layers except the first and the last ones. The activation function is a Leaky ReLU to mitigate gradient vanishing. The output function is a sigmoid function. Adam [31] is used as an optimizer since its based on a stochastic gradient descent method that is lightweight in terms of computation and deals well with noisy or sparse gradients. The cost function is the binary cross entropy loss function. The model was implemented using Python 3 and Keras with Tensorflow 2.4.0.

The model expects 4 RSSI values as input, obtained by 4 consecutive packets. These packets should be sampled one packet per second with as even spacing as possible. This was done to decouple the data rate from the algorithm. Thus, it is possible to sample from the packet stream in cases with higher data rate and keep the model lightweight to complete the computation before the model is scheduled to run again.

Fig. 6 presents the two buffers we used to label our data. The first buffer represents the packets that have arrived and has a size of 4 packets, the fourth packet with the arrow represents the packet received most recently. This is the “pre” buffer. Then the prediction or the “post” buffer has a size of 7 packets. Missing values in the input data were replaced with the value of -100 dB representing a very weak signal. Each RSSI value in the post buffer is checked and if any value is below the threshold (which is -87 dBm), we label the pre-buffer data as 1, meaning that they will produce a disconnect event or as 0 if there is no value below the threshold meaning that the connection is maintained. This process is the same for all the classes of data. The example in Fig. 6 illustrates that the data in pre buffer will produce a disconnect event since packets with RSSI below the threshold follow in the post buffer.

Based on the data classification we train three different models, one with the artificial data, one with the enhanced artificial data and another one with the real world data. For training purposes we divided each class of data to 75% for training and 15% for validating. For evaluation, all models were tested on 10% of the real world data. As shown in Fig. 7, our datasets suffer from class imbalance. Following previous works [32], we adopt weights in the loss function

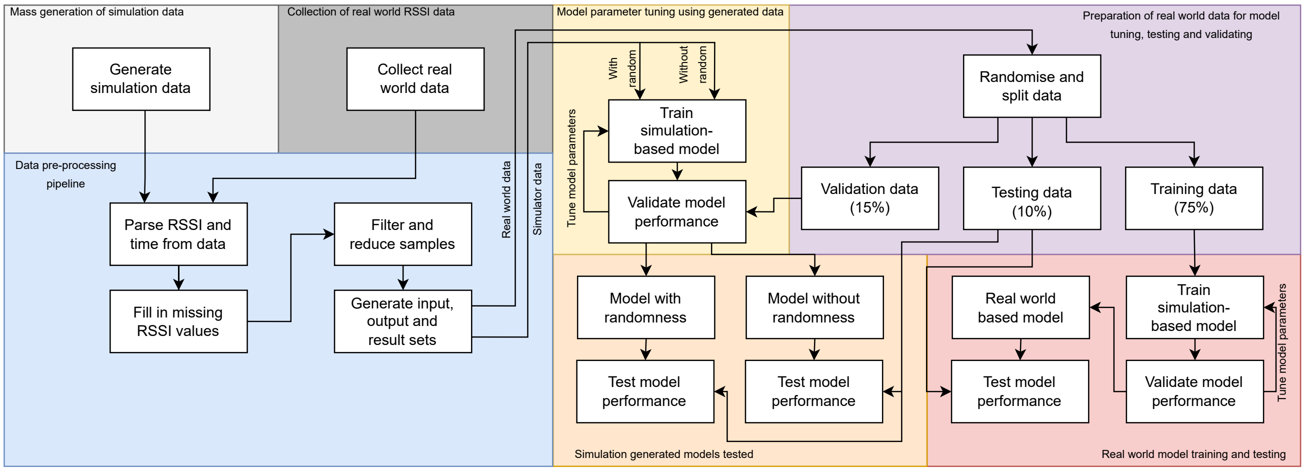

Fig. 4. The pipeline from generating and collecting data, pre-processing, training, testing, to the finished model.

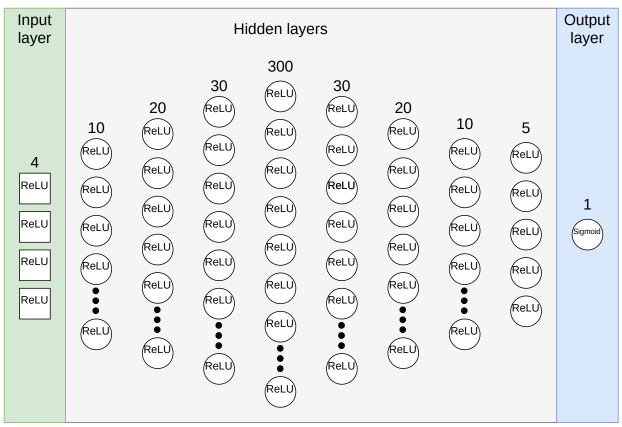

Fig. 5. The architecture of the Neural Network, all the nodes are fully connected using the activation function specified in each node.

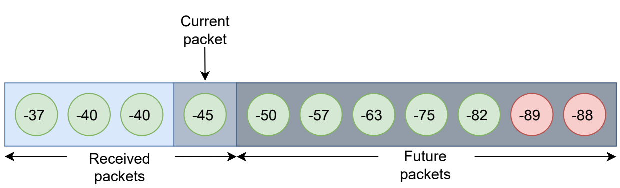

Fig. 6. The pre and post buffers used to label the data. In this particular example the received packets in the pre buffer will be labeled as 1 meaning that they will produce a disconnect event because there are two packets bellow the threshold in the post buffer.

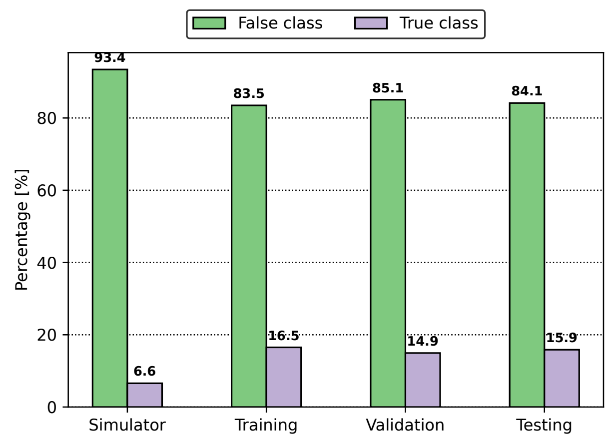

Fig. 7. Class imbalance among different data sets used to train and evaluate the model.

to further penalize misclassifications of the underrepresented class.

V. Model Evaluation

Table I lists the confusion matrices of the three models we evaluate. More specifically the *Artificial Model* refers to the model generated from artificial data, the *Enhanced Model* to the model generated from the artificial data enhanced with real measurements and the *Real-World Model* to the model generated from real-world data. Since the output from the model is a confidence value, it was decided to tune the models to have a similar percentage of False Positives (FPs) and True Positives (TPs). This was done since the main goal of our use case is to capture as many cases of the disconnect events as possible.

TABLE I
Confusion Matrices

Artificial Model		Ground Truth	
		Disconnection	No Disconnection
Prediction	Disconnection	74.65%	51.15%
	No Disconnection	25.35%	48.85%

Enhanced Model		Ground Truth	
		Disconnection	No Disconnection
Prediction	Disconnection	84.98%	45.66%
	No Disconnection	15.02%	54.34%

Real-World Model		Ground Truth	
		Disconnection	No Disconnection
Prediction	Disconnection	86.38%	30.27%
	No Disconnection	13.62%	69.73%

Observing the results at Table I we see that the perfor-

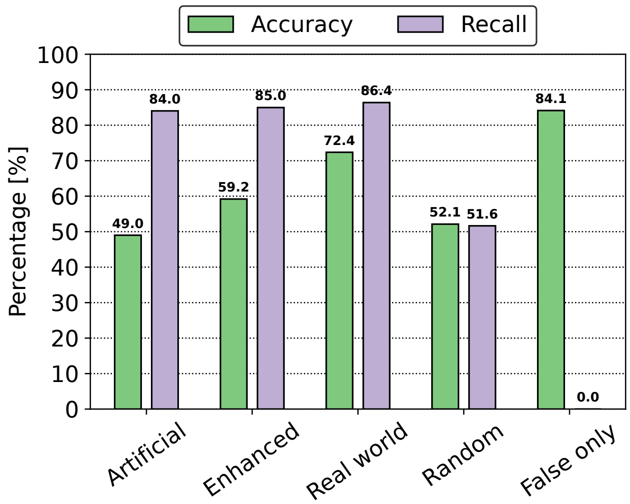

Fig. 8. Accuracy and recall of all models including a random and a false only models.

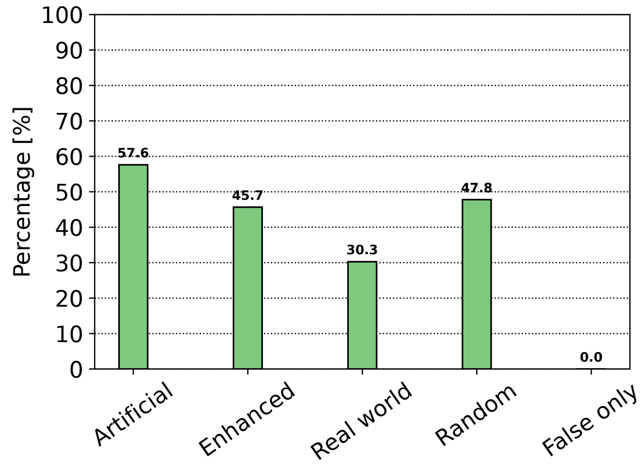

Fig. 9. False alarm rate of all models.

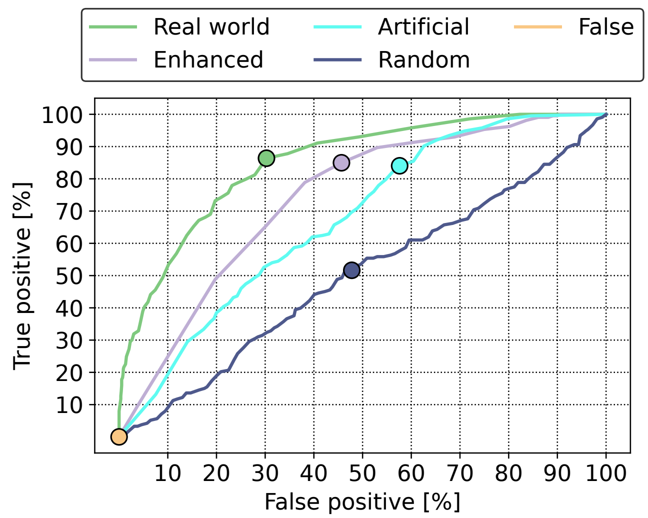

Fig. 10. ROC curves for all models including the benchmark models. The markers represent the threshold values used during testing.

mance is improved when we enhanced the artificial data by introducing some noise (*Enhanced Model*). Specifically the reduction of FPs from 51.15% down to 45.46% and False Negatives (FNs) from 25.35% to 15.02%. The reduction in FNs is very important in a safety critical case, as missing a disconnecting event might be crucial. The reduction in FPs does reduce the noise produced by the system from false alarms. We observe also that the *Real-World Model* has the best performance.

For further evaluation, Fig. 8 includes two benchmark models, one that predicts with 50% probability between the false and true class (*Random*) and one that predicts just the false class (*False Only*). The *False Only* model scores the highest accuracy since there is a heavy imbalance in favour of the false class. The model with the best accuracy after the false only is the real world model with 72.4% accuracy. The low accuracy can be explained by the large number FPs. The accuracy can be increased but that would assume an increase on the FNs and the goal is to minimize FNs with an acceptable level of FPs. Regarding the recall metric, which does not taking into account the FPs, the three models have similar scores as expected since they were tuned to get a similar number of TPs. The highest score is marginally in favour of the *Real-World Model* which with 86.3%. The recall for the *Random Model* is 52.6% which is in line with randomly guessing while the *False Only Model* is 0%. Fig. 9 depicts the false alarm rate and the *Real-World Model* again presents the lowest rate with 30.3%.

The Receiver Operating Characteristics (ROC) curve for all the models is presented in Fig. 10. The threshold values selected are 0.2 for the *Real-World Model*, 0.17 for the *Enhanced Model*, 0.15 for the *Artificial Model* and 0.5 for the *Random Model* and the *False Only Model*. Those values result in the models giving a similar number of TPs while the TNs differ between the models. The only model without a curve is the *False Only Model* which is expected as the true positive rate and the false positive rate is 0% for all threshold values. The curve for the *Random Model* follows an almost straight diagonal line which is expected if the random function uniformly distributes the values. By moving the threshold up, the model stops only predicting one class and starts predicting both classes until the maximum threshold value is reached again where it starts only predicting the other class resulting in the maximum true positive rate and false positive rate.

The threshold value chosen for the *Real-World Model* is close to optimal for maximising TPs while still letting as few FPs. The same goes for both the other models but they perform worse in the majority. If the threshold is changed to 0.4 the accuracy rises to 86.22% due to the increase rate of TNs which is higher than the decrease rate of the TPs. This is again due to the imbalance of the classes in the dataset.

There are indeed much fewer FNs but when we look at the recall it drops down to 38.97% which is a significant decrease. If the threshold is lowered on the other hand the opposite happens. When the threshold is at 0.1 the recall is 100% but the accuracy is only at 26.58% due to most of the classifications being FPs.

VI. Conclusion

Various autonomous vehicles applications, such as fleets of connected autonomous agricultural vehicles, have safety requirements that deem necessary the avoidance of wireless disconnection events. In this paper, we present a framework for predicting that a disconnection event is about to happen seconds before it actually happens. The idea is that this prediction shall initiate a proactive action, such as adjusting the velocity or trajectory of the vehicle, to prevent the disconnection from happening. In this context, we collect data and train three models: one on data obtained from a simulator, one on real-world data, and one the simulator data augmented with real-world noise. The results demonstrate that the model trained on augmented data outperforms the model trained on simulated data, yielding performance comparable to the model trained on real-world data. This is particularly valuable in scenarios where collecting real-world disconnection events is costly or even impossible due to regulatory requirements.

Acknowledgements

This work was partially funded by Innovation Fund Denmark, grant 9092-00007B AgroRobottiFleet.

References

[1] A. A. Brincat, F. Pacifici, S. Martinaglia, and F. Mazzola, "The Internet of Things for Intelligent Transportation Systems in Real Smart Cities Scenarios," in *IEEE 5th World Forum on Internet of Things (WF-IoT)*, pp. 128–132, 2019.

[2] E. Sisinni, A. Saifullah, S. Han, U. Jennehag, and M. Gidlund, "Industrial Internet of Things: Challenges, Opportunities, and Directions," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 11, pp. 4724–4734, 2018.

[3] O. Elijah, T. A. Rahman, I. Orikumhi, C. Y. Leow, and M. N. Hindia, "An overview of internet of things (iot) and data analytics in agriculture: Benefits and challenges," *IEEE Internet of Things Journal*, vol. 5, no. 5, pp. 3758–3773, 2018.

[4] C. Lytridis, V. G. Kaburlasos, T. Pachidis, M. Manios, E. Vrochidou, T. Kalampokas, and S. Chatzistamatis, "An overview of cooperative robotics in agriculture," *Agronomy*, vol. 11, no. 9, 2021.

[5] P. Simoens, M. Dragone, and A. Saffiotti, "The Internet of Robotic Things: A review of concept, added value and applications," *International Journal of Advanced Robotic Systems*, vol. 15, no. 1, 2018.

[6] A. Kamilaris and N. Botteghi, "The Penetration of Internet of Things in Robotics: Towards a Web of Robotic Things," *J. Ambient Intell. Smart Environ.*, vol. 12, p. 491–512, jan 2020.

[7] H. Wang, H. Zhao, J. Zhang, D. Ma, J. Li, and J. Wei, "Survey on unmanned aerial vehicle networks: A cyber physical system perspective," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 2, pp. 1027–1070, 2020.

[8] S. Duquennoy, A. Elsts, B. A. Nahas, and G. Oikonomou, "TSCH and 6TiSCH for Contiki: Challenges, Design and Evaluation," in *2017 13th International Conference on Distributed Computing in Sensor Systems (DCOSS)*, pp. 11–18, 2017.

[9] C. Orfanidis, A. Elsts, P. Pop, and X. Fafoutis, "TSCH Evaluation Under Heterogeneous Mobile Scenarios," *IoT*, vol. 2, no. 4, pp. 656–668, 2021.

[10] S. Raza, T. v. d. Lee, G. Exarchakos, and M. Güneş, "A Reliability Analysis of TSCH Protocol in a Mobile Scenario," in *2019 16th IEEE Annual Consumer Communications & Networking Conference (CCNC)*, pp. 1–6, 2019.

[11] M. Jayasuriya, R. Ranasinghe, and G. Dissanayake, "Active Perception for Outdoor Localisation with an Omnidirectional Camera," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4567–4574, IEEE, 2020.

[12] L. Marchegiani and P. Newman, "Learning to listen to your ego-(motion): Metric motion estimation from auditory signals," in *Annual Conference Towards Autonomous Robotic Systems*, pp. 247–259, Springer, 2018.

[13] O. Bartlett, C. Gurau, L. Marchegiani, and I. Posner, "Enabling intelligent energy management for robots using publicly available maps," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2224–2229, 2016.

[14] P. Ondrúška, C. Gurău, L. Marchegiani, C. H. Tong, and I. Posner, "Scheduled perception for energy-efficient path following," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4799–4806, 2015.

[15] N. Baccour, A. Koubâa, L. Mottola, M. A. Zúñiga, H. Youssef, C. A. Boano, and M. Alves, "Radio Link Quality Estimation in Wireless Sensor Networks: A Survey," *ACM Trans. Sen. Netw.*, sep 2012.

[16] K. Srinivasan and P. A. Levis, "RSSI is Under Appreciated," in *3rd Workshop Embedded Networked Systems (EmNets)*, 2006.

[17] N. Baccour, A. Koubâa, M. Ben Jamâa, H. Youssef, M. Zuniga, and M. Alves, "A comparative simulation study of link quality estimators in wireless sensor networks," in *2009 IEEE International Symposium on Modeling, Analysis & Simulation of Computer and Telecommunication Systems*, pp. 1–10, 2009.

[18] Y. Chen and A. Terzis, "On the mechanisms and effects of calibrating rssi measurements for 802.15.4 radios," in *Wireless Sensor Networks* (J. S. Silva, B. Krishnamachari, and F. Boavida, eds.), (Berlin, Heidelberg), pp. 256–271, Springer Berlin Heidelberg, 2010.

[19] M. Senel, K. Chintalapudi, D. Lal, A. Keshavarzian, and E. J. Coyle, "A kalman filter based link quality estimation scheme for wireless sensor networks," in *IEEE Global Telecommunications Conference (GLOBECOM)*, pp. 875–880, 2007.

[20] K. Srinivasan, P. Dutta, A. Tavakoli, and P. Levis, "An Empirical Study of Low-Power Wireless," *ACM Trans. Sen. Netw.*, mar 2010.

[21] K. Farkas, T. Hossmann, F. Legendre, B. Plattner, and S. K. Das, "Link quality prediction in mesh networks," *Computer Communications*, vol. 31, no. 8, pp. 1497–1512, 2008.

[22] T. Liu and A. E. Cerpa, "Foresee (4c): Wireless link prediction using link features," in *Proceedings of the 10th ACM/IEEE International Conference on Information Processing in Sensor etworks*, 2011.

[23] T. Liu and A. E. Cerpa, "Talent: Temporal adaptive link estimator with no training," in *Proceedings of the 10th ACM Conference on Embedded Network Sensor Systems*, SenSys '12, (New York, NY, USA), p. 253–266, Association for Computing Machinery, 2012.

[24] T. Liu and A. E. Cerpa, "Temporal adaptive link quality prediction with online learning," *ACM Trans. Sen. Netw.*, vol. 10, may 2014.

[25] J. Webber, N. Suga, S. Ano, Y. Hou, A. Mehbodniya, T. Higashimori, K. Yano, and Y. Suzuki, "Machine learning-based rssi prediction in factory environments," in *2019 25th Asia-Pacific Conference on Communications (APCC)*, pp. 195–200, 2019.

[26] F. Osterlind, A. Dunkels, J. Eriksson, N. Finne, and T. Voigt, "Cross-Level Sensor Network Simulation with COOJA," in *31st IEEE Conference on Local Computer Networks*, pp. 641–648, 2006.

[27] Texas Instruments, "SimpleLink™ CC2650 wireless MCU LaunchPad™ Development Kit," 2019.

[28] T. Szandała, "Review and Comparison of Commonly Used Activation Functions for Deep Neural Networks," *ArXiv*, vol. abs/2010.09458, 2020.

[29] S. A. Alameri and M. Mohd, "Comparison of Fake News Detection using Machine Learning and Deep Learning Techniques," *2021 3rd International Cyber Resilience Conference (CRC)*, pp. 1–6, 2021.

[30] T. Liu and A. E. Cerpa, "Data-Driven Link Quality Prediction Using Link Features," *ACM Trans. Sen. Netw.*, vol. 10, no. 2, 2014.

[31] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," *CoRR*, vol. abs/1412.6980, 2014.

[32] Z. Wu, Y. Guo, W. Lin, S. Yu, and Y. Ji, "A weighted deep representation learning model for imbalanced fault diagnosis in cyber-physical systems," *Sensors*, vol. 18, no. 4, 2018.