A Fuzzy-Based Machine Learning Model for Robot Prediction of Link Quality

Christopher J. Lowrance
Department of Electrical Engineering and Computer Science
United States Military Academy
West Point, NY USA
Christopher.Lowrance@usma.edu

Adrian P. Lauf, Mehmed Kantardzic
Department of Computer Engineering and Computer Science
University of Louisville
Louisville, KY USA
Adrian.Lauf@louisville.edu, Mmkant01@louisville.edu

***Abstract*—With foresight into the state of the wireless channel, a robot can make various optimization decisions with regards to routing packets, planning mobility paths, or switching between diverse radios. However, the process of predicting link quality (LQ) is nontrivial due to the streaming and dynamic nature of radio wave propagation, which is complicated by robot mobility. Due to robot movement, the wireless propagation environment can change considerably in terms of distance, obstacles, noise, and interference. Therefore, LQ must be learned and regularly updated while the robot is online. However, the existing fuzzy-based models for assessing LQ are non-adaptable due to the absence of any learning mechanism. To address this issue, we introduce a fuzzy-based prediction model designed for the efficient online and incremental learning of LQ. The unique approach uses fuzzy logic to infer LQ based on the collective output from a series of offset classifiers and their posterior probabilities. In essence, the proposed model leverages machine learning for extracting the underlying functional relationship between the input and output variables, but deeper inferences are made from the output of the learning algorithms using fuzzy logic. Wireless link data from a real-world robot network was used to compare the model with the traditional linear regression approach. The results show statistically significant improvements in three out of the six real-world indoor and outdoor environments where the robot operated. Additionally, the novel approach offers a number of other benefits, including the flexibility to use fuzzy logic for model tuning, as well as the ability to make implementation efficiencies in terms of parallelization and the conservation of labeling resources.**

I. Introduction

Robots typically transmit a steady stream of data, consisting of sensor readings and video imagery, to other nodes, such as an observation control unit [1]. The challenge is that a robot transmitter is natively unaware of the link quality (e.g., throughput) at the receiver without explicit feedback, and without this information from the receiver, a robot may inadvertently cause its wireless connectivity to degrade to an insufficient quality as it moves. Unfortunately, having the destination node consistently transmit feedback across the wireless channel to the sender is relatively expensive in terms of energy and frequency-channel utilization. A more efficient alternative would be to have the sending robot perform LQ prediction in lieu of the receiver providing explicit feedback after every packet reception. In other words, the transmitting robot would use a set of relatively inexpensive inputs, such as received signal strength indicator (RSSI) from its radio, and relate them to the desired target of LQ using machine learning.

With the ability to gain link-awareness in a cost-effective manner, a robot can realistically exploit the knowledge in a variety of ways. For instance, as a robot moves, it could make LQ predictions and build a spatial mapping of its connectivity area using the historical set of predictions and feedback examples [2]. With a spatial awareness of LQ, a robot can plan its movement in an attempt to optimize network connectivity [3]. The concept of planning a mobility path that is favorable for wireless connectivity is essential for robots serving as mobile relays [4], or those collaborating and moving with team members [5-7]. The ability to accurately foresee LQ prior to each packet transmission can also be incorporated into the ad-hoc networking protocols of robots, so that the most reliable and efficient routes are selected for packet transmissions [8, 9]. Finally, predictions on LQ are essential for the timely handoff between diverse radio systems, as described in [10, 11], where backup directional radio systems are activated by robots based on LQ to extend the range of communications.

Unfortunately, it is difficult for a robot to predict in advance how well its impending transmission will be received at the destination. One of the challenges is that there are few low-cost metrics that can serve as predictors of LQ [12]. Furthermore, the metrics provided by the radio hardware tend to fluctuate rapidly as a result of being directly related to the effects of radio wave propagation [13]. The duration that these physical-layer metrics remain stationary (i.e., the channel coherence time) can be on the order of microseconds in some mobile networks [14]. However, robots, like other higher-layer applications, often cannot complete optimization adjustments that quickly. As a result, the radio metrics must be statistically related to LQ in a generalized fashion, and therefore, the metrics can be considered somewhat noisy. Another complication is that LQ is dependent upon the characteristics of the radio hardware (e.g., antenna gain, transmit power, etc.) and the propagation environment (e.g., amount of noise, interference, and shadowing characteristics) [15]. Therefore, concept drift or a deviation in the statistical relationship between the input metrics and the output LQ indicator can occur whenever these environmental factors change considerably.

Because the wireless channel is dynamic in mobile networks, a robot must learn the dependencies between the input features and the target variable while operating online. Furthermore, the robot must incrementally update its prediction model over time in order to mitigate the effects of concept drift. To meet these objectives, this paper presents a fuzzy-based prediction model capability of adaptation through supervised learning and batch-style retraining. The model uses a series of classifiers to statistically relate the inputs at the sender to an indicator of LQ measured at the receiver. The fuzzy framework is incorporated through the strategic labeling perspective of these classifiers, and the way the collective outputs from the classifiers are interpreted in a fuzzy manner to infer a continuous (i.e., non-discrete) measure of LQ.

The fuzzy-based prediction model is unique and offers several advantages. Similar to other fuzzy-based systems, the model is relatively straightforward to modify in terms of tuning or incorporating domain knowledge because its inference mechanism is based on human reasoning and described linguistically. However, unlike the existing fuzzy-based LQ models, the architecture presented in this paper is adaptable because of its internal classifiers and supervised learning framework. Another advantage is that the task of learning from examples is divided among a series of binary classifiers, which can be programmed to execute concurrently in order to gain speedup. The speed and efficiency of the prediction model is an important aspect given the streaming nature of LQ and the resource constraints of most robots.

Another novelty of the proposed model is its batch-style learning format, which contrasts with the existing LQ prediction models that make tuning adjustments based on individual samples. Learning in batches can reduce labeling costs and improve model stability. In terms of overhead reduction, the energy and network costs of labeling is not insignificant, and therefore, the knowledge provided by a particular sample should be leveraged as long as possible until it should be eventually forgotten due to concept drift. On the other hand, the existing models are memoryless systems because they dispose of any sample knowledge immediately after the model is tuned with it [16], and this intuitively seems wasteful given the cost of labeling. As for model stability, the wireless channel can produce samples that are outliers due to the effects of deep fading [14]. Consequently, prediction adjustments based on single samples, such as those that have undergone deep fades, can have adverse effects on model accuracy. To mitigate this issue, the proposed model learns in batches so that changes are based on a statistical generalization over multiple samples.

The precise makeup of a batch-style framework that adaptively samples the environment and incrementally presents new samples to the proposed model is left for future work. Instead, the primary purpose of this paper is to take the preliminary step of detailing the fuzzy-based machine learning method and to evaluate its accuracy in an *offline* manner using *real-world* datasets. These datasets, which were extracted from an actual robot network, are available online to facilitate further research in the field.

The remainder of the paper is organized as follows. Section II provides an overview of the related work. Subsequently, the proposed fuzzy-based model is described in Section III. In Section IV, the model is evaluated using real datasets from a mobile ad-hoc network. Finally, Section V provides concluding remarks and future work proposals.

II. Related Work

A. Fuzzy-based Methods for Predicting LQ

Empirical studies have concluded that LQ cannot be accurately assessed using a single input metric [12, 15], and therefore, multiple metrics should be considered to improve accuracy. Fuzzy logic has been leveraged in some works for combining multiple inputs to estimate LQ in wireless sensor networks (WSNs) [17-19]. However, these fuzzy-based models quantify LQ using fuzzy sets that were configured offline based on prior experimentation. Furthermore, the fuzzy sets of these designs remain static while the system is online under the assumption that the overlapping regions of the fuzzy sets will cope with any unexpected variations in the input metrics.

In the domain of fuzzy control [20], there exist adaptive models that perform system identification and tracking. However, the control paradigm is inappropriate for LQ prediction given that most applications have little or no control over the input features. Outside of the controls domain, there exists fuzzy systems that can learn from their environment. However, these fuzzy-based learners are generally intended to automate the process of fuzzy rule-generation through mechanisms such as neural networks [21]. In contrast, this work is focused on adapting to concept drift by learning the mapping relationships (i.e., fuzzy memberships) relating the inputs variables to the target output.

The closest resemblance to the proposed approach outside the domain of LQ estimation is fuzzy cluster analysis [22]. Both methods share the fundamental concept of assigning levels of certainty to class memberships using posterior probabilities, but the problem domain, as well as the execution details, are different as will become more evident in Section III.

B. Machine Learning Methods for Predicting LQ

Some works utilize machine learning algorithms based on time series analysis for LQ prediction [23-26]. Liu et al. use a weighted sum of ordered past observations of packet reception ratio (PRR) to forecast its future value [23]. However, the statistical mapping of two radio metrics to PRR was performed offline. In another study, Farkas et al. employs pattern matching via cross correlation of time series data to predict LQ [24, 25]. This approach would be difficult and resource-intensive based on the wide variety of pattern possibilities. Another work by Millan et al. uses time-series analysis, but the datasets were extracted from a stationary network with arguably less variation than mobile networks [26].

Caleffi and Paura employ neural networks to predict the expected number of transmissions (ETX) to deliver the next packet [27, 28]. The neural-based predictor showed promising results in simulation. However, in a follow-up study with Cacciapuoti et al., the performance of the neuron estimator was inconclusive [27]. Another possible concern with the neural-based approach would be its computational requirements compared to other machine learning techniques [29].

The use of regression techniques in the form of supervised learning have also been explored in a series of works [29-34]. In [29, 31], a distributed protocol was designed to exploit the mobility of nodes for gathering diverse training samples, but the models were only trained once and not designed to be incremental learners. Di Caro et al. developed an online learning framework that incrementally retrains its regression model [30] using Locally Weighted Projection Regression (LWPR), but the evaluations were based on datasets from a stationary WSN. Wang et al. [34] evaluated the use of classifiers for quantifying LQ into discrete categories such as *'good'* or *'bad'*, but the system offers limited granularity. Liu and Cerpa studied the use of machine learning for predicting the chances of successful packet delivery over a short-term window [32, 33]. The works primarily differ in that the learning took place offline in [33], while the model was updated online in [32].

In general, the aforementioned works utilized existing algorithms, which are often available in online repositories such as [35, 36]. The novelty of these works existed in the application of different algorithms, features, or target variables within the domain of LQ prediction. This paper differs in that it leverages existing machine learning algorithms to develop an alternative model that is based on fuzzy logic.

III. Architectures of the Fuzzy-Based Learner

A. The Mamdani Variant of the ETF Model

Fuzzy systems generally perform a sequence of main subprocesses, consisting of fuzzification, inference, and defuzzification, to generate crisp (i.e., continuous) outputs from multiple input variables [20]. Traditionally, the fuzzification step is accomplished using triangular-shaped, Gaussian-shaped, or other types of functions that are often setup using domain knowledge and frequently remain static after configuration. However, given the dynamic nature of the wireless channel, membership functions need to be adaptable due to concept drift. To incorporate adaptability, the proposed model leverages a series of binary classifiers to assign the input samples into fuzzy sets (i.e., linguistic values) with a level of certainty based on the posterior probabilities of the classifiers. The classifiers adapt their decision boundaries based on the new samples incorporated into each new training batch.

The first version of the proposed model only modifies the traditional step of fuzzification, where inputs are mapped into fuzzy sets. Hence, the remaining logic of inference and defuzzification using fuzzy sets, which is the trademark of Mamdani systems, remains intact. The usual means of fuzzification is replaced in the proposed model using a series of binary classifiers that are diversely emplaced with respect to the target variable as depicted in Fig. 1. The figure shows a series of received signal strength indicator (RSSI) samples plotted with respect to the target label of throughput potential ratio (TPR). TPR is defined as

$$TPR_i = \frac{r_i}{max(r)} \tag{1}$$

where r_i is the data rate for the i^{th} transmission and $max(r)$ is the highest data rate observed since link inception. TPR is only measured and accurately known to the receiver, whereas RSSI is readily available for sampling from the sender's radio.

The collection of classifiers, indicated by the three horizontal and colored lines in Fig. 1, forms a type of ensemble. As a result, the overall approach is referred to as the *ensemble-to-fuzzy* (ETF) model. It is worth noting that the use of multiple classifiers within the ETF model does not meet the usual definition of an ensemble. Traditionally, the term *ensemble* refers to a diverse and independent set of predictive models that operate on the same training data, and the results of which are then combined in some fashion to generate a collective prediction that is consistently more accurate than any individual model [36]. On the other hand, the ensemble used in the ETF model consists of classifiers that use the *same* learning algorithm, but they operate on *different* class labels due to their unique placement on the target universe.

The first step in setting up the ETF model is to divide the target universe into the desired number of equally-spaced regions or fuzzy sets. Each fuzzy set is then logically assigned a linguistic value. As shown on the left side of Fig. 1, the sample space was classified into four different fuzzy sets that correspond to linguistic throughput potential ratio (TPR) levels of *'high'* (H), *'moderate high'* (MH), *'moderate low'* (ML), and *'low'* (L). The number of binary classifiers needed to make the fuzzy distinctions is one less than the number of selected fuzzy sets.

Once the labeling boundaries of the fuzzy sets have been established, training examples must be collected and assigned unique class labels for each individual classifier. As indicated in Fig. 1, each classifier within the ensemble labels a sample with a binary '1' if the example resides above its target threshold; otherwise, a binary '0' would be assigned. After labeling sufficient examples, the classifiers are trained using the *same* type of supervised learning algorithm, but each classifier uses its own set of *independent* labels.

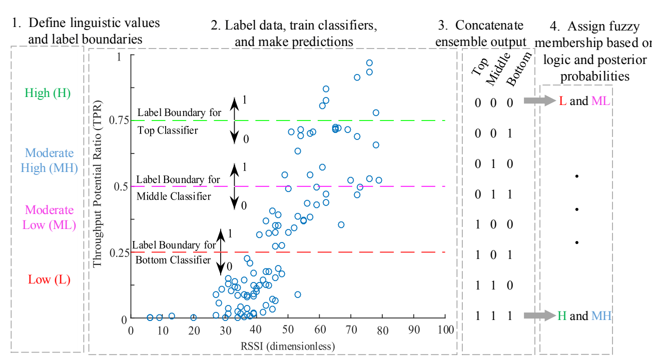

Fig. 1. Steps for adaptable fuzzification using binary classifiers.

After training, the ensemble can proceed to assign fuzzy memberships to any unlabeled sample it is presented. The process starts by concatenating together the classification decisions from each classifier. The concatenated output forms a discrete set of binary combinations as indicated in Fig. 1. Together, the outputs identify the two most appropriate fuzzy sets to which a sample belongs. Fuzzy reasoning and *if-then* logic are used to establish these assignments based on the level of agreement exhibited within the ensemble. For instance, in the case of an ensemble output of '000', all of the classifiers agree that the sample primarily belongs to the fuzzy set of *'low'* (L), but in the essence of fuzzy logic, the sample may also partially belong with a degree of less than 0.5 to the neighboring (i.e., secondary) fuzzy set of *'moderate low'* (ML). In this example case of '000', the precise membership levels to each neighboring fuzzy set would be assigned using the posterior probabilities from the bottom classifier because it divides the two fuzzy sets where the sample should logically reside.

Table I specifies the fuzzy membership assignments for every possible ensemble output of Fig. 1. The input variable, x, in the posterior probabilities of $P(0|x)$ and $P(1|x)$ in Table I is annotated as a single variable for this section, but the input variable should be treated as a vector of features, **x**, in the next section. For any particular case in Table I, a sample is only assigned membership into two fuzzy sets. Furthermore, only a single classifier performs membership assignments using its $P(0|x)$ and $P(1|x)$, and therefore, the sum of the fuzzy memberships is one. Fuzzy sets not listed for a particular case in Table I are set to zero.

Besides the straightforward cases of '000' and '111', the other possibilities outlined in Table I require some additional consideration before selecting the two most appropriate fuzzy sets. For instance, sub-nested *if-then* logic shown in Table I is used in the cases of '001' and '011' in order to identify the most-fitting secondary fuzzy set for a given sample. As an example, consider the ensemble output of '011'; in this case, the classifiers agree that the sample primarily belongs to the *'moderate high'* (MH) fuzzy set. However, it is unclear whether the secondary fuzzy set should be *'high'* (H) or *'moderate low'* (ML) because the sample may, in actuality, lie either toward the top or middle classifier boundary. To make this determination, the two nearest classifiers are ranked with regard to their posterior probabilities. Specifically, in this example case, if the middle classifier is more confident about its primary classification than the top classifier, then it logically implies that the sample resides closer to the top boundary and the *'high'* (H) fuzzy set; otherwise, it implies the sample resides more toward the middle boundary and the *'moderate high'* (MH) fuzzy set. Finally, there are some cases, such as '010' and '101', where there is no adjacent agreement between the classifiers and majority-rule logic does not make sense. In the event of these rare cases, the best option is to make the fuzzy assignment decision using the middle classifier and to assign the sample into the two moderate fuzzy sets as shown in Table I.

TABLE I
FUZZIFICATION LOGIC USED IN THE ETF METHOD

Ensemble Output Top-Middle-Bottom	**Fuzzy Set Assignments**
0-0-0	$L = P_{bottom}(0\|x), ML = P_{bottom}(1\|x)$
0-0-1	if $P_{bottom}(1\|x) > P_{middle}(0\|x)$, then $ML = P_{mid}(0\|x), MH = P_{mid}(1\|x)$ else $ML = P_{bottom}(1\|x), L = P_{bottom}(0\|x)$
0-1-0	$ML = P_{mid}(0\|x), MH = P_{mid}(1\|x)$
0-1-1	if $P_{middle}(1\|x) > P_{top}(0\|x)$, then $MH = P_{top}(0\|x), H = P_{top}(1\|x)$ else $MH = P_{mid}(1\|x), ML = P_{mid}(0\|x)$
1-0-0	$ML = P_{bottom}(1\|x), L = P_{bottom}(0\|x)$
1-0-1	$ML = P_{mid}(0\|x), MH = P_{mid}(1\|x)$
1-1-0	$MH = P_{top}(0\|x), H = P_{top}(1\|x)$
1-1-1	$MH = P_{top}(0\|x), H = P_{top}(1\|x)$

Once each feature has been fuzzified using the above approach, the traditional fuzzy steps of inference and defuzzification would commence in order to generate a non-discrete prediction. The fuzzy-based structure of the proposed model easily allows for tuning of the output to potentially improve accuracy. For instance, one possibility could consist of adding more fuzzy sets by introducing more classifiers within the ensemble. This approach could lead to more fidelity in sample fuzzification and ultimately better accuracy. Additionally, the rule-base logic outlined in Table I could be modified based on domain knowledge.

B. The Tagaki-Sugeno Variant of the ETF Model

Normally in fuzzy logic, input variables are fuzzified independently. Likewise, the Mamdani version of the ETF model processes the inputs individually, which requires the maintenance of multiple ensembles. A more streamlined approach would be allow the classifiers in the ETF model to operate on a vector of features during classification; this approach would only require a single ensemble. Therefore, the i^{th} training example, $\langle \mathbf{x}_i, y_i \rangle$, in the modified version of the ETF model would consist of a combined feature vector $\mathbf{x_i}$ and a single target label, y_i.

Another challenge with traditional Mamdani fuzzy architectures is the so-called *combinatorial explosion problem* [38]. As the number of inputs and fuzzy sets increase, the system and its rule base become more complex. The modified version of the ETF model does not inherit this problem because the steps of inference and defuzzification are also performed using the ensemble output.

Similar to the previous version of the ETF architecture, the binary predictions from each classifier are concatenated and then processed through rule-base (i.e., *if-then*) logic. However, instead of only using the posterior probabilities to infer to fuzzy set memberships, this version of the ETF directly infers a crisp output based on the ensemble output. This modified version is said to be inspired by the fuzzy architecture of Tagaki-Sugeno (T-S) because an algebraic expression is used to describe the consequent of the ensemble output.

The example in Fig. 2 was created to assist with the explanation of the T-S variant. The figure only shows the target universe and omits any particular input universe because the feature space is assumed to be multidimensional. The concatenated output of the ensemble in the figure example is shown to be '001'. Based on this ensemble output, the classifiers agree that the sample primarily belongs somewhere in the *'moderate low'* fuzzy set. Thus, the sample originates in the center or midpoint of that particular fuzzy set. However, in reality, the sample may lie somewhere above or below the midpoint. Therefore, in order to more accurately predict a value for that sample, the posterior probabilities of select classifiers are leveraged as indicated in Fig. 2. Using logic similar to that of Table I, the sample may be assigned a value slightly lower or higher than the midpoint depending upon a ranking of posterior probabilities.

Table II formalizes the membership assignments for the remaining output possibilities of the T-S variant, assuming the ensemble consists of three classifiers as shown in Fig. 2. The outputs annotated with an asterisk correspond to significant disagreement within the ensemble and would rarely, if ever, occur. Regardless, if any of these rare cases occur, the rule base logic conservatively assigns the sample to the middle fuzzy sets of MH and ML.

The primary and secondary fuzzy set assignments outlined in Table II are transformed into an output prediction that is crisp (i.e., continuous) in nature using the following algebraic equation

$$y^{crisp} = m_j \pm \frac{1}{n}\mu_k \quad (2)$$

where m_j is the target midpoint value associated with the primary fuzzy set, j, and μ_k is the level of membership to secondary fuzzy set, k. The constant n is the number of fuzzy sets used to divide the target universe, and it is responsible for bounding the maximum amount of shift the predicted output can assume away from the midpoint, m_j. Furthermore, the final output prediction is restricted to the extreme boundaries of primary fuzzy set based on the $1/n$ term and the fact that the membership to the secondary set, μ_k, will always be less than 0.5. The assignment of plus (+) or minus (-) sign in the above equation corresponds to whether the shift from the midpoint value is higher or lower, respectively, and the sign is determined by the rules established within the *if-then* logic.

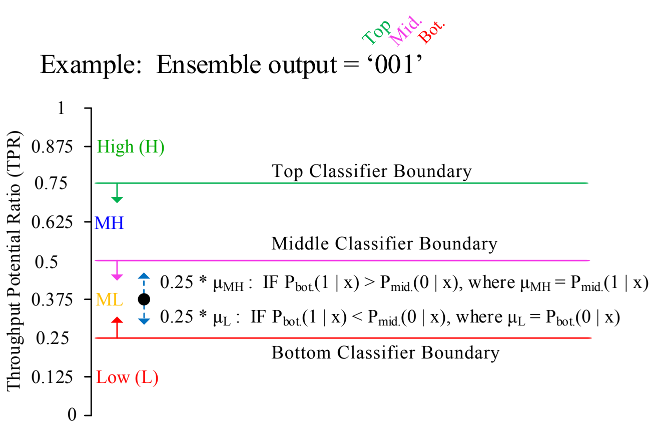

Fig. 2. An example to illustrate the inference mechanism used in Takagi-Sugeno ETF architecture.

TABLE II
FUZZY SET ASSIGNMENTS OF T-S VARIANT

Ensemble Output Top-Mid-Bot	**Primary Fuzzy Set** (sets m_j)	**Secondary Fuzzy Set** (μ_k)
0-0-0	L	$ML = P_{bot}(1\|x)$
0-0-1	ML	if $P_{bot}(1\|x) > P_{mid}(0\|x)$ then $MH = P_{mid}(1\|x)$ else $L = P_{bot}(0\|x)$
0-1-0*	L	$ML = P_{bot}(1\|x)$
0-1-1	MH	if $P_{mid}(1\|x) > P_{top}(0\|x)$ then $H = P_{top}(1\|x)$ else $ML = P_{mid}(0\|x)$
1-0-0*	ML	$MH = P_{mid}(1\|x)$
1-0-1*	ML	$MH = P_{mid}(1\|x)$
1-1-0*	MH	$ML = P_{mid}(0\|x)$
1-1-1	H	$MH = P_{top}(0\|x)$

IV. REAL-WORLD VALIDATION: LEARNING LINK QUALITY IN A ROBOT NETWORK

A. Experimental Setup of the Robot Networks

The proposed model was evaluated in an offline manner using a series of real-world datasets from a robot network that operated in different environments. The setup of each experiment was similar to the photograph shown in Fig. 3, where the robot was tele-operated by a laptop observation control unit (OCU) over an IEEE 802.11 ad hoc connection. With the exception of the indoor experiment, the robot started near the OCU and was commanded to move farther away in a linear fashion until the wireless link eventually failed. This route of travel was selected in order to force the robot to collect measurements ranging from very good to very poor LQ.

During the duration of each experiment, the robot transmitted a picture image, along with sensor data, approximately every 1.5 seconds to the OCU based on the imaging speed of the camera and other factors. The typical payload of each packet ranged from about 60 kilobytes up to 1 megabyte depending upon the size of the picture image. Each packet was transmitted via Transmission Control Protocol (TCP). Immediately prior to each transmission, the robot sampled a vector of LQ features; these features included the radio hardware metrics of RSSI and signal quality (SQ) extracted from the Linux */proc* filesystem, the statistic known as the

Fig. 3. Picture showing the robot and OCU collecting LQ data along the sidewalk of a residential neighborhood.

expected number of transmissions (ETX) formed using bi-directional probes which were transmitted every second [39], and the distance to the OCU calculated using global positioning system (GPS) coordinates. The OCU recorded the throughput of each packet it received and then transformed the target variable into a throughput potential ratio (TPR), which was defined in (1).

A total of six experiments were conducted at various locations, including inside and outside of a residential building, a community park, and a recreational track, as listed in Table III. At some locations, the *line-of-sight* (LOS) between the radio antennas was temporarily obstructed, and therefore, these datasets are labeled with NLOS for *non-line-of-sight*. All of the datasets are available online at [40] to easily facilitate future work. In addition, the same online location [40] includes charts showing the characteristics of the features with respect to TPR, as well as videos showing the data collection process using the robot and OCU.

B. Classifier and Feature Selection for the ETF Model

Before evaluating the ETF model, a preliminary step is to identify the optimal performing classifier algorithm for the application. The following classifiers were evaluated during the selection process: Naïve Bayes, Support Vector Machine (SVM), and Logistic Regression. The open-source repository of machine learning algorithms known as *scikit-learn* [35] was used during the classifier testing, as well as for the ensemble employed in the ETF model.

TABLE III
DESCRIPTION OF SAMPLE COLLECTION SITES

Dataset Name	Sample Size	Description
Residential (Outdoor)	400	Robot moved along sidewalk surrounded by homes
Residential (Indoor)	429	Robot and OCU on separate floors of residential building
Park (LOS)	415	Robot moved through parking lot free of obstacles
Park (NLOS)	377	An automobile was placed between robot and OCU for portion of dataset
Track (LOS)	417	Robot moved along side of track with homes on side
Track (NLOS)	161	A box was placed over robot for a portion of dataset

Cross validation (CV) results revealed that the SVM and Logistic Regression algorithms both outperformed the Naïve Bayes and had similar accuracies. However, the speed tests on the robotic platform, which used an ARM 700 MHz processor [41], revealed that the logistic regression algorithm was able to train its prediction model significantly faster than the SVM (i.e., nearly 400 ms faster for a training set of 400 samples). Based on these results, logistic regression was selected as the classifier algorithm to be used in further testing of the ETF.

Recursive feature elimination (RFE) was used to evaluate the value of each feature in terms of aiding the predictive model. The two radio hardware metrics of SQ and RSSI were closely matched and ranked the first and second most informative, respectively. Distance ranked third and ETX followed. All features were retained in further tested based on each of them occasionally proving to be the most informative during RFE testing, and only marginal training and prediction time differences were noted between the three and four feature options.

C. Comparing Accuracy of ETF to Linear Regression

The complete ETF system produces an output that is continuous in nature, similar to a regression-style learning algorithm. Therefore, the predictive accuracy of ETF was compared to the most commonly employed regression method, linear regression (LR) [37]. The algorithm implementation of LR was drawn from *scikit-learn* [35].

In terms of the ETF model, the T-S variant was used during the evaluations based on its improved scalability over the Mamdani version detailed in Section III. Both ETF and LR were evaluated using the aforementioned datasets and 10-fold CV. For each individual test case, ETF and LR generated LQ predictions in a pairwise fashion. The mean absolute error (MAE) from each prediction within every fold was recorded in a list for each algorithm.

The list of MAEs produced by each algorithm was then compared statistically after all 10 folds were complete. Boxplots were initially used, as shown in Fig. 4, to visually inspect some of the key statistical attributes about the errors generated by each algorithm. The boxplots show the median value of the ETF model to be lower on three of the six datasets. Furthermore, the upper whiskers of the ETF model are shown to be lower on four of the experiments. Additionally, the two different models produced roughly the same error statistics on the residential datasets based on the medians and quartile edges of the boxes being nearly the same. Finally, Fig. 4 shows that linear regression algorithm produced the most extreme outliers on five of the six different datasets.

To more precisely quantify the mean difference (μ_d) between the models, paired t-tests were performed using the pairwise MAEs from all of the datasets. The results of the

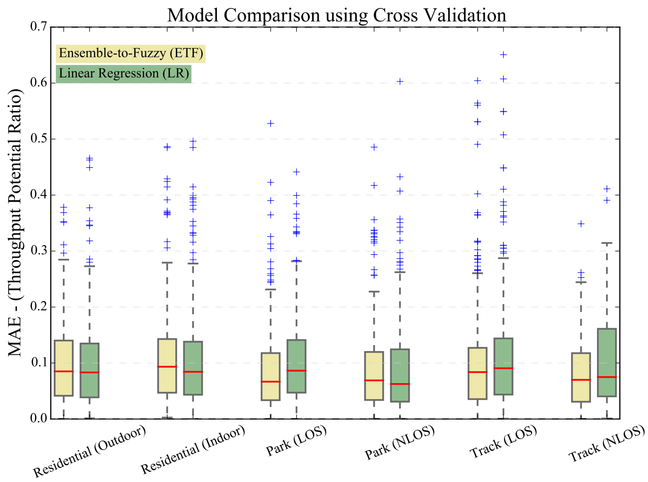

Fig. 4. Boxplots comparing the mean absolute error generated by the ETF and LR algorithms during 10-fold CV testing of various datasets.

paired t-tests on each dataset are displayed in Table IV. As the table indicates, the null hypothesis (i.e., $\mu_d = 0$) was rejected on four of the datasets, meaning that the predictive accuracies of the models were statistically different on four of the datasets. On three of the four datasets that were statistically different, the ETF model proved to be slightly more accurate. On the other hand, the accuracy of the two algorithms did not prove statistically different on two datasets.

In summary, the ETF model was shown to generally outperform LR on the given datasets. For instance, the mean prediction error calculated across all six datasets was roughly 10% for LR and 9% for ETF. Furthermore, the LR model failed to surpass the ETF model by more than a percentage point on any of the given datasets. Whereas, the ETF model surpassed LR by a mean of up to 2.5% on two of the datasets [e.g., see Table IV for Park (LOS) and Track (NLOS)] and more than 1% on another [e.g., see Table IV Track (LOS)]. It is worth noting that pushing the average error much lower than 10%, as the ETF model did, is a significantly difficult task. Generally speaking, it is common that machine learning algorithms to achieve 90% accuracy on a wide range of applications [37], but exceeding that accuracy becomes exponentially more challenging.

TABLE IV
ACCURACY DIFFERENCES BASED ON PAIRED T-TESTS

Dataset	**Null Hypothesis** $\mu_d = 0$ at 5% Signif.	**Confidence Interval** (LR - ETF)
Residential (Outdoor)	Not Rejected	-0.0051< μ_d <0.0019
Residential (Indoor)	Rejected	-0.0071< μ_d <-0.0007
Park (LOS)	Rejected	0.0132< μ_d <0.0244
Park (NLOS)	Not Rejected	-0.0047< μ_d <0.0036
Track (LOS)	Rejected	0.0064< μ_d <0.0144
Track (NLOS)	Rejected	0.0080< μ_d <0.0269

TABLE V
TRAINING AND PREDICTION SPEEDS OF THE ETF MODEL

Portion of Algorithm	**Time (ms)**
Training ETF (1 Classifier)	29.33
Training ETF (3 Classifiers)	89.00
Training LR	13.93
Prediction ETF (1 Classifier)	3.85
Prediction ETF (3 Classifiers)	11.72
Prediction LR	2.82

D. *Evaluating Model Implementation Efficiency*

The training and prediction speed of the ETF model was evaluated on the robot's 700 MHz single-core ARM processor [41]. In order to baseline the execution times, the speed of LR was also evaluated. The execution time averages of Table V were calculated using every training and prediction event of the 10-fold CV testing for the outdoor residential dataset. The prediction algorithms were implemented in Python using *scikit-learn* [35].

As shown in Table V, the ETF model was evaluated in terms of the training and prediction speed of one classifier, as well as the sequential execution of the complete ensemble of three classifiers. The purpose the distinction was to show the potential speedup of the model if the individual classifiers were programmed to execute in parallel across a multicore system. The table also shows that the ETF model requires more computational time than LR, but as mentioned in Section I, the fuzzy interface of the ETF method offers a series of advantages in terms of its intuitiveness and flexibility for tuning. Furthermore, it should be noted that LR is one of the most lightweight, yet effective prediction models [37]; therefore, most other prediction models incur additional computations beyond LR.

As expected, the process of training the model is more computationally-intense than prediction. However, given the batch learning framework, retraining events would transpire less frequently than generating predictions. In summary, Table V demonstrates that both training and prediction tasks of the ETF model can be feasibly executed online using the aforementioned robot, given the camera speed and the approximate 1.5 second window between transmissions.

V. CONCLUSIONS AND FUTURE WORK

This paper introduced a fuzzy-based machine learning method for predicting LQ in robot networks. We evaluated our approach by comparing the model's prediction results with those of traditional linear regression. The results show statistically significant improvement on the majority of the experiments conducted using a real robot in a variety of environments. The datasets from these robot experiments are available online to the community for further research. A significant advantage of the proposed model is that it offers enhanced flexibility in terms of model tuning or adding domain knowledge because of its overarching fuzzy design.

There are several opportunities for future work related to the ETF model. The next evolution of this work is to evaluate the ETF model in an online setting, where samples are presented to the model in a streaming fashion. The evaluation presented in this paper used nearly complete datasets, consisting of examples ranging from good to poor LQ. However, in a streaming environment, training examples are not presented all at once, and it would be interested to evaluate the robustness of the ETF under these conditions. Furthermore, evaluating the ETF in a stream-based setting would also allow for an investigation into the efficiency of its batch-style model when compared to single-instance learners. Finally, it would be insightful to apply the ETF model to other domains outside of LQ prediction, and whether the flexibility of its fuzzy framework can be leveraged to outperform existing methods in other problem areas.

REFERENCES

[1] H. G. Nguyen, N. Pezeshkian, A. Hart, A. Burmeister, K. Holz, J. Neff, et al., "Evolution of a radio communication relay system," in *Proc. SPIE 8741*, Unmanned Systems Technology XV, 2013.

[2] J. N. Twigg, J. Fink, P. L. Yu, and B. M. Sadler, "Efficient Base Station Connectivity Area Discovery," in *The International Journal of Robotics Research*, July 30, 2013.

[3] E. F. Flushing, M. Kudelski, L. M. Gambardella, and G. A. Di Caro, "Spatial prediction of wireless links and its application to the path control of mobile robots," in *IEEE Int. Symposium on Industrial Embedded Systems*, 2014, pp. 218-227.

[4] B. Degener, S. P. Fekete, B. Kempkes, and F. M. Auf Der Heide, "A survey on relay placement with runtime and approximation guarantees," in *Computer Science Review*, vol. 5, pp. 57-68, 2011.

[5] J. Fink, A. Ribeiro, and V. Kumar, "Robust control for mobility and wireless communication in cyberphysical systems with application to robot teams," *Proceedings of the IEEE*, vol. 100, pp. 164-178, 2012.

[6] M. A. Hsieh, A. Cowley, V. Kumar, and C. J. Taylor, "Maintaining network connectivity and performance in robot teams," in *Journal of Field Robotics*, vol. 25, pp. 111-131, 2008.

[7] P. Cruz, R. Fierro, W. Lu, and S. Ferrari, "Maintaining robust connectivity in heterogeneous robotic networks," in *Proc. SPIE 8741, Unmanned Systems Technology XV*, 2013.

[8] M. Campista, P. Esposito, I. Moraes, et al., "Routing Metrics and Protocols for Wireless Mesh Networks," in *IEEE Network*, vol. 22, pp. 6-12, 2008.

[9] J. Zhou, M. Jacobsson, E. Onur, and I. Niemegeers, "An Investigation of Link Quality Assessment for Mobile Multi-hop and Multi-rate Wireless Networks," in *Wireless Personal Communications*, vol. 65, pp. 405-423, 2012.

[10] C. J. Lowrance and A. P. Lauf, "Adding transmission diversity to unmanned systems through radio switching and directivity," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp. 3788-3793, 2014.

[11] C. J. Lowrance and A. P. Lauf, "Link Estimation in Robot Networks for Multi-Radio Control and Range Extension," in *Journal of Intelligent & Robotic Systems*, pp. 1-31, 2016.

[12] A. Vlavianos, L. K. Law, I. Broustis, S. V. Krishnamurthy, and M. Faloutsos, "Assessing link quality in IEEE 802.11 Wireless Networks: Which is the right metric?," in *IEEE 19th International Symposium on Personal, Indoor and Mobile Radio Communications*, pp. 1-6, 2008.

[13] C. Renner, S. Ernst, C. Weyer, and V. Turau, "Prediction accuracy of link-quality estimators," in *Wireless Sensor Networks*, ed: Springer, 2011, pp. 1-16.

[14] A. Goldsmith, *Wireless Communications*. Cambridge University Press, 2005.

[15] N. Baccour, A. Koubaa, L. Mottola, M. A. Zuniga, H. Youssef, C. A. Boano, et al., "Radio link quality estimation in wireless sensor networks: a survey," in *ACM Transactions on Sensor Networks*, vol. 8, p. 34, 2012.

[16] M. A. Maloof and R. S. Michalski, "Incremental learning with partial instance memory," Artificial intelligence, vol. 154, pp. 95-126, 2004.

[17] N. Baccour, A. Koubaa, H. Youssef, et al., "F-LQE: A Fuzzy Link Quality Estimator for Wireless Sensor Networks," in *Wireless Sensor Networks*, vol. 5970, ed: Springer Berlin Heidelberg, 2010, pp. 240-255.

[18] J. Ko and M. Chang, "MoMoRo: Providing Mobility Support for Low-Power Wireless Applications," in *IEEE Systems Journal*, vol. 9, pp. 585-594, 2014.

[19] G. Zhi-Qiang, W. Qin, L. Mo-Han, and H. Jie, "Fuzzy Logic Based Multidimensional Link Quality Estimation for Multi-Hop Wireless Sensor Networks," *IEEE Sensors Journal*, vol. 13, pp. 3605-3615, 2013.

[20] J. H. Lilly, *Fuzzy control and identification*. Hoboken, N.J.: Wiley, 2010.

[21] O. Cordón, "A historical review of evolutionary learning methods for Mamdani-type fuzzy rule-based systems: Designing interpretable genetic fuzzy systems," in *Int. Journal of Approximate Reasoning*, vol. 52, pp. 894-913, 2011.

[22] E. Hüllermeier, "Fuzzy sets in machine learning and data mining," in *Applied Soft Computing*, vol. 11, pp. 1493-1505, 2011.

[23] L. Liu, Y. Fan, J. Shu, and K. Yu, "A link quality prediction mechanism for wsns based on time series model," in *7th Int. Conf. on Ubiquitous Intelligence & Computing and Autonomic & Trusted Computing*, 2010, pp. 175-179.

[24] K. Farkas, T. Hossmann, F. Legendre, B. Plattner, and S. K. Das, "Link quality prediction in mesh networks," in *Computer Communications*, vol. 31, pp. 1497-1512, 2008.

[25] K. Farkas, T. Hossmann, L. Ruf, and B. Plattner, "Pattern matching based link quality prediction in wireless mobile ad hoc networks," in *ACM Int. Symposium on Modeling Analysis and Simulation of Wireless and Mobile Systems*, 2006, pp. 239-246.

[26] P. Millan, C. Molina, E. Medina, D. Vega, R. Meseguer, B. Braem, et al., "Tracking and predicting link quality in wireless community networks," in *IEEE 10th Int. Conf. on Wireless and Mobile Computing, Networking and Communications*, 2014, pp. 239-244.

[27] A. S. Cacciapuoti, M. Caleffi, L. Paura, and M. Rahman, "Link quality estimators for multi-hop mesh network," in *Euro Med Telco Conference*, 2014, pp. 1-6.

[28] M. Caleffi and L. Paura, "Bio-inspired link quality estimation for wireless mesh networks," in *IEEE Int. Symposium on a World of Wireless, Mobile and Multimedia Networks & Workshops*, 2009, pp. 1-6.

[29] M. Kudelski, L. M. Gambardella, and G. A. Di Caro, "A mobility-controlled link quality learning protocol for multi-robot coordination tasks," in *IEEE Int. Conf. on Robotics and Automation*, pp. 5024-5031, 2014.

[30] G. A. Di Caro, M. Kudelski, E. F. Flushing, J. Nagi, I. Ahmed, and L. M. Gambardella, "Online supervised incremental learning of link quality estimates in wireless networks," in *IEEE Mediterranean Ad Hoc Networking Workshop*, 2013, pp. 133-140.

[31] E. F. Flushing, J. Nagi, and G. A. Di Caro, "A mobility-assisted protocol for supervised learning of link quality estimates in wireless networks," in *Int. Conf. on Computing, Networking and Communications*, 2012, pp. 137-143.

[32] T. Liu and A. E. Cerpa, "Temporal Adaptive Link Quality Prediction with Online Learning," in *ACM Transactions on Sensor Networks*, vol. 10, p. 46, 2014.

[33] T. Liu and A. E. Cerpa, "Data-driven link quality prediction using link features," in *ACM Transactions on Sensor Networks*, vol. 10, p. 37, 2014.

[34] Y. Wang, M. Martonosi, and L.-S. Peh, "Predicting link quality using supervised learning in wireless sensor networks," in *ACM Mobile Computing and Communications Review*, vol. 11, pp. 71-83, 2007.

[35] scikit-learn: Machine Learning in Python [Online]. Available: http://scikit-learn.org/stable/

[36] Weka 3: Data Mining Software in Java [Online]. Available: http://www.cs.waikato.ac.nz/ml/weka/

[37] M. Kantardzic, *Data Mining: Concepts, Models, Methods, and Algorithms*: John Wiley & Sons, 2011.

[38] W. E. Combs and J. E. Andrews, "Combinatorial rule explosion eliminated by a fuzzy rule configuration," in *IEEE Transactions on Fuzzy Systems*, vol. 6, pp. 1-11, 199 8.

[39] D. S. De Couto, D. Aguayo, J. Bicket, and R. Morris, "A high-throughput path metric for multi-hop wireless routing," in *Wireless Networks*, vol. 11, pp. 419-434, 2005.

[40] Datasets. Available: https://louisville.edu/speed/computer/research/aerial-robotics-lab

[41] Broadcom. BCM2835. Available: http://www.broadcom.com/products/BCM2835