Shapley Value Approximation via Machine Learning in Cooperative Games for Traffic Equilibrium*

Mauro Passacantando† Giorgio Gnecco‡ Marcello Sanguineti§

Abstract. Estimating the importance of arcs and nodes in a network is a major issue in a variety of applications, from movement analysis to traffic management and wildfire fighting. This can be done by exploiting the Shapley value, a concept from cooperative games with transferable utility, as a measure of the significance of each player in such games. To this end, one can define games in which the players are network elements (e.g., arcs or nodes). However, for large networks, the exact evaluation of the Shapley value is computationally demanding. Here, an approach for its approximate computation is investigated. The network is parameterized by one or more quantities of interest (e.g., the traffic demand). Smoothness properties of the Shapley value as a function of such parameters are investigated and exploited to apply machine learning techniques for its approximate computation. The methodology is tested on two networks: the one on which the so-called Braess' paradox (i.e., adding a resource may in some cases deteriorate, instead of improving, the overall network performance) was first observed and the Sioux-Falls network, a benchmark for network analysis and design, having realistic topology and demands. Numerical results show the effectiveness of the proposed machine learning approximations of the Shapley value.

Keywords. Shapley Value, Machine Learning, Approximate Computation, Transferable-Utility Games, Wardrop Equilibrium.

2020 MSC: 91A12, 68T07.

1 Introduction

A major issue in network analysis consists of defining and measuring the "importance" of its nodes/arcs. This can help decision-makers, e.g., in detecting weak network components, identifying and preventing failures, controlling fire outbreaks in extensive wilderness areas, and, in general, improving network connectivity (for instance, in terms of access, costs, flow, reliability, and travel time).

To quantify the relative importance of nodes, a possible approach is based on the concept of *centrality* [29]. Among classical measures of node centrality, *degree centrality* is based on the concept of node degree, computed starting from the information on the network encoded by its adjacency matrix. *Closeness centrality* of a node i corresponds to the inverse of the sum of the lengths of the shortest paths between i and all the other nodes. *Betweenness centrality* is defined in terms of the comparison of the number of all the

*This paper has been published in Journal of Dynamics and Games, DOI: 10.3934/jdg.2025038, see: https://www.aimsciences.org/article/doi/10.3934/jdg.2025038

†University of Milano-Bicocca, Milan, Italy

‡IMT School for Advanced Studies, Lucca, Italy

§University of Genova, Italy, National Research Council of Italy, Genova, Italy, corresponding author.

shortest paths between any pair of other nodes that include (do not include, resp.) the node i. *Eigenvalue centrality* relies on the idea that the importance of i is related to its connection to nodes with a high score.

However, classical measures of node centrality present several drawbacks. For instance, they do not consider how a network behaves if any subset of nodes is removed. Moreover, typically they do not consider the *Origin-Destination (OD)* demand. Finally, some relevant transportation-related network properties (e.g., arc capacity, route choice, and traffic congestion) are not easily captured (especially jointly) by such classical measures.

Alternative measures of node centrality are based on *solution concepts* coming from *cooperative game theory* [31] and can deal with such drawbacks. Another advantage is that, differently from most classical measures of node centrality, game-theoretical measures of node centrality can be extended in a straightforward way to evaluate the centrality of the network arcs[1]. Moreover, game-theoretical measures of arc centrality may take negative values for some arcs, even when all the arc costs are non-negative [34]. This is relevant from a network design perspective since it may be convenient to remove such arcs from the network. Applications of game-theoretical measures of centrality range, e.g., from movement analysis [19] to traffic networks [34] and wildfire fighting [1, 39].

In this work, we introduce one among such game-theoretical measures of *arc centrality*, which is based on a solution concept from cooperative game theory, the *Shapley value* [42], and we investigate its advantages and limitations. The proposed model is a cooperative game with *transferable utility*, where the players are a subset of arcs in the network and the utility is defined in terms of user equilibrium.

We discuss computational issues related to the evaluation of the Shapley value in the game and investigate its smoothness properties with respect to parameters of interest (i.e., the traffic demand). Such properties are exploited to estimate the accuracy of its computation via *Machine Learning (ML)* techniques. This motivates the application of ML methods for its approximation.

Two case studies are considered to illustrate the proposed approach via extensive numerical results. The first refers to the Braess' network, in which the so-called Braess' paradox (according to which including one additional resource in a network may, in some cases deteriorate, instead of improving, its overall performance) was first discovered. The second case study is related to the Sioux-Falls network. This is a medium-sized network having realistic topology (with 24 nodes and 76 arcs) and realistic demands, which is often used as a benchmark for network analysis and design.

Preliminary results were presented at a conference [13]. In particular, Sections 2, 3, 4, 5.1, and 7 are improved and expanded versions of the corresponding sections in [13], whereas Sections 5.2 and 6 are completely novel[2].

The paper is organized as follows. Section 2 contains some basic definitions and concepts related to cooperative games with Transferable Utility (TU games) and introduces the family of Transportation Network cooperative games (TNc games), which provide the theoretical framework of the analysis. In Section 3, the TNc game based on the user equilibrium is introduced. Section 4 is devoted to the use of ML for Shapley value approximation. In Sections 5 and 6, numerical results on the Braess' network and the Sioux-Falls network are presented, respectively. Finally, Section 7 is a discussion.

[1] However, a variation of betweenness centrality (called *edge betweenness centrality* [12]) holds for edges.

[2] From the numerical point of view, the conference paper [13]: does not present any result for the two-dimensional version of Case Study 1 (see Section 5.2 of the present manuscript); does not consider training with a noisy training set (generated either by the Monte Carlo or the Kernel Shap method) in the case of the real-world network investigated here in the Case Study 2 (see Section 6); considers only the application of one ML model (the Artificial Neural Network model, ANN), not the ε-Support Vector Regression model (ε-SVR; Section 5.1); does not compare the results in terms of the normalized Kendall τ ranking distance (see again Section 5.1 of the present manuscript).

2 Preliminaries: cooperative games on networks

2.1 Cooperative games with transferable utility

A *cooperative game with Transferable Utility*, or *TU game*, is a pair (N, v) defined in the following way: N represents a set of players (this set is also called *grand coalition*); each subset $S \subseteq N$ is called a (*sub*)*coalition*. The real-valued mapping $v : 2^N \to \mathbb{R}$ (in which 2^N is the power set of N, i.e., the set of all its possible subsets), is named *characteristic function* (or *utility function*). It assigns to each coalition S the value $v(S)$. This represents the utility that can be obtained jointly by all the players belonging to S when they do not receive any contribution from the other players (i.e., those belonging to $N \setminus S$). By definition, for the *empty coalition* (i.e., the one containing no players), one has $v(\emptyset) = 0$. The quantity $v(N)$ represents the total utility of the grand coalition N, whereas, for every player i, $v(\{i\})$ is the utility achieved by player i alone.

In TU games, each player can transfer part of its utility to any other player without incurring any loss (e.g., through a common "currency", valued in the same way by each player). TU games can be investigated and solved by relying on suitable *solution concepts* of *cooperative game theory*. Each of them represents a criterion for dividing the total utility $v(N)$ of the grand coalition among the individual players. The higher the amount of the total utility assigned to one player by a solution concept, the higher the importance of that player in the grand coalition. In this work, we adopt as a solution concept the *Shapley value*, which is defined as follows:

$$Sh(i) = \sum_{S \subseteq N} \frac{(|S|-1)!(|N|-|S|)!}{|N|!} [v(S) - v(S \setminus \{i\})], \qquad \forall\, i \in N. \tag{1}$$

In the above, $v(S) - v(S \setminus \{i\})$ represents the marginal utility[3] of player i when one moves from the coalition $S \setminus \{i\}$ to the coalition S. Hence, according to Equation (1), the Shapley value $Sh(i)$ of player i represents the expected marginal utility of that player, when the expectation is computed across all the possible coalitions, and the players, starting from the empty coalition, enter sequentially the grand coalition in a random order of appearance (in such a way that all the $|N|!$ orders of appearance have the same probability $1/|N|!$). This interpretation of the Shapley value allows one to adopt this solution concept as a measure of the importance of the players even in frameworks in which such players cannot be assumed to be rational decision-makers. This occurs, for instance, in the case of features in games associated with supervised ML problems [8], and in the case of genes in the so-called microarray games [26].

Finally, we recall here several properties of TU games that will be considered in some parts of the work (see, e.g., [7, 38] for further discussions on such properties). A TU game (N, v) is called

- *subadditive* when $v(S \cup T) \leq v(S) + v(T)$ is valid for any pair of disjoint coalitions $S, T \subseteq N$;
- *superadditive* when $v(S \cup T) \geq v(S) + v(T)$ is valid for any pair of disjoint coalitions $S, T \subseteq N$;
- *monotonic* when $v(S) \leq v(T)$ is valid for any $S, T \subseteq N$ for which $S \subset T$;
- *convex* when $v(S \cup T) + v(S \cap T) \geq v(S) + v(T)$ is valid for any $S, T \subseteq N$;
- *cohesive* when $v(N) \geq \sum_{i=1}^{k} v(S_i)$ is valid for any partition $\{S_1, \ldots, S_k\}$ of N.

It follows from the definitions above that any convex game is also superadditive, whereas any superadditive game is also cohesive.

[3] In this article, one defines the marginal utility of each player i as $v(S) - v(S \setminus \{i\})$, i.e., with respect to a coalition S from which that player is removed (if initially present, otherwise the two coalitions S and $S \setminus \{i\}$ are identical); equivalently, this marginal utility could have been defined as $v(S \cup \{i\}) - v(S)$, i.e., with respect to a coalition S to which player i is added (if initially absent, otherwise the two coalitions $S \cup \{i\}$ and S are identical).

2.2 Transportation network cooperative games (TNc games)

Let a directed graph $G = (V, A)$ be given, where V is a set of nodes, and $A \subseteq V \times V$ is a set of arcs. In the game-theoretical approach adopted in this work, the set N of players is modeled as a subset of (critical) arcs (i.e., arcs on which the analysis is focused). The directed graph associated with a generic coalition $S \subseteq N$ is denoted by

$$G(S) := (V,\ S \cup (A \setminus N)). \tag{2}$$

In other words, the directed graph $G(S)$ has the same set of nodes as the original directed graph G, it contains the arcs of the coalition S, and all the arcs that do not belong to the player set N of the game (N, v). In this framework, nodes can represent, e.g., critical transportation components, intersections, points of interest, stations, stops, and transportation centroids. Similarly, arcs can represent, e.g., physical connections between pairs of nodes, such as entire routes, parts of routes, rails, and roads.

The characteristic function can be defined in such a way as to model a broad range of transportation features, e.g., assigned demand, connectivity, distance, and travel time-saving. For instance, in [17], various *Transportation Network cooperative games* (or *TNc games*) were defined, by modeling the characteristic function of each TU game taking into account the TN topology and some network attributes[4]. Then, each TU game was solved in terms of the Shapley values of all the players, represented therein by the nodes of the network. The resulting Shapley values were exploited to evaluate the centrality of each node in the network. Theoretical properties of this measure of centrality were analyzed in [14, 17]. However, a drawback of the characteristic functions considered in [17] is that they are not able to model traffic congestion. Indeed, they refer to cases in which the cost of every arc is given in an exogenous way. In this article, instead, we are interested in taking into account traffic congestion to model such costs. Moreover, we consider the arcs as players (instead of the nodes). As detailed in the next section, the utility of each coalition can be defined based on the equilibrium flows arising in a related traffic assignment problem [37].

3 The TNc game based on the user equilibrium

In the article, we adopt the following notations and definitions related to equilibrium flows. First, let W denote a set of *Origin-Destination (OD) pairs*, i.e., a subset of ordered pairs of nodes of G. The *demand* associated with an OD pair $w \in W$ is denoted by d_w, whereas the *vector of demands* is represented by d. A *path* (i.e., a sequence of nodes $v_1, \ldots, v_k \in V$ such that for each i from 1 to $k-1$, one has $(v_i, v_{i+1}) \in A$) is denoted by p, whereas the non-negative *flow on the path* p in G (which can be thought as a uniform non-negative quantity traversing the sequence of all the arcs of p) is represented by x_p, and the *vector of path flows* by x. Analogously, the flow on the path p in $G(S)$ is denoted by $x_p(S)$, and the vector of all the path flows in $G(S)$ by $x(S)$.

The non-negative *flow on the arc* i is represented by f_i, and the *vector of all the arc flows* by f. The set of *acyclic paths* (i.e., paths in which no vertex is repeated) joining the OD pair w is denoted by P_w, whereas the set of all the paths that join the OD pair w in $G(S)$ is represented by $P_w(S)$, and the set of all the paths that join the OD pairs in G (resp., $G(S)$) is denoted by P (resp., $P(S)$). The (non-negative) *cost* on the arc i related to the flow vector f (i.e., the time needed to traverse that arc) is represented by $c_i(f)$. The *incidence arc-path matrix* is denoted by Δ. Its elements are defined as follows: $\Delta_{i,p} := 1$ when arc i belongs to path p, 0 otherwise. To conclude, the *user cost* on the path p in G (resp., $G(S)$), which is the sum of the costs on all the arcs of such path, is represented by $C_p(x)$ (resp. $C_p(x(S))$), whereas the vector of the costs on all the paths in G (resp., $G(S)$) is denoted by $C(x)$ (resp., $C(x(S))$).

[4] In [17], undirected graphs were considered, with no substantial change in the definition of a TNc game.

The vector of arc flows f and the vector of path flows x are related as follows: $f_i = \sum_{p \in P} \Delta_{i,p} x_p$. A path flow x is defined as *feasible* if it satisfies all the demands, i.e., if $\sum_{p \in P_w} x_p = d_w$ for any OD pair $w \in W$.

In the article, a characteristic function based on the user equilibrium is considered, based on the following work hypothesis: all the drivers have perfect knowledge of the travel costs associated with all the arcs of the network, and select the best route based on the so-called *Wardrop's first principle* [45], according to which "no driver can unilaterally reduce his/her travel cost by shifting to another route" (see also the concept of *Nash equilibrium* [28]). In this way, a deterministic user equilibrium is achieved.

In the following, we define $\mathcal{F}$ as the set of coalitions $S \subseteq N$ for which, given any OD pair $w \in W$, there exists at least one path in $G(S)$ that connects w. For any coalition $S \in \mathcal{F}$, a *Wardrop equilibrium*, or *User Equilibrium (UE)*, in the network $G(S)$ is defined as any feasible path flow vector $\bar{x}(S)$ for which, given any OD pair $w \in W$ and any $p \in P_w(S)$, one has

$$C_p(\bar{x}(S)) \begin{cases} = \lambda_w(S) & \text{if } \bar{x}_p(S) > 0, \\ \geq \lambda_w(S) & \text{if } \bar{x}_p(S) = 0, \end{cases} \tag{3}$$

where $\lambda_w(S)$ can be interpreted as the "equilibrium disutility" for the OD pair w. In other words, both $\bar{x}(S)$ and the vector $\lambda(S)$ are implicitly defined as solutions to the set of all the equilibrium conditions of the form (3) obtained by varying $w \in W$ and $p \in P_w(S)$. Morover, it can be proven that looking for a Wardrop equilibrium $\bar{x}(S)$ reduces to solving the following *variational inequality* [10, 43]: find $\bar{x}(S)$ such that

$$\langle C(\bar{x}(S)), x(S) - \bar{x}(S) \rangle \geq 0, \qquad \text{for any feasible path flow } x(S). \tag{4}$$

Once $\bar{x}(S)$ is found[5] by solving problem (4), its corresponding disutility vector $\lambda(S)$ is obtained directly from conditions (3), by looking at the paths $p \in P_w(S)$ for which $\bar{x}_p(S) > 0$ (there is always at least one such path, due to the feasibility of $\bar{x}(S)$). Finally, the Total Cost (or total travel time) associated with the Wardrop equilibrium $\bar{x}(S)$ is defined as

$$TC^{ue}(\bar{x}(S)) = \sum_{w \in W} \sum_{p \in P_w(S)} \bar{x}_p(S) C_p(\bar{x}(S)). \tag{5}$$

It is worth noting that, when multiple Wardrop equilibria $\bar{x}(S)$ exist, this total travel time does not depend on the specific choice of one among these Wardrop equilibria if, for each coalition $S \in \mathcal{F}$, the arc equilibrium flow is unique. This occurs, e.g., when the costs of the arcs are continuous, separable, and increasing functions of the respective flows[6][9, 37]. Under this assumption on the costs of the arcs (which is made in the rest of the work)[7], $TC^{ue}(\bar{x}(S))$ depends only on S – hence, with a slight abuse of notation, it can be denoted as $TC^{ue}(S)$ – and its values obtained by varying the coalition $S \in \mathcal{F}$ define the total cost function $TC^{ue} : \mathcal{F} \to \mathbb{R}$.

Definition 3.1. Based on the total cost function $TC^{ue} : \mathcal{F} \to \mathbb{R}$, we define the characteristic function $v^{ue} : 2^N \to \mathbb{R}$ by considering the following two cases:

(a) When the empty coalition $\emptyset \in \mathcal{F}$ (i.e., all the coalitions are in $\mathcal{F}$), one sets

$$v^{ue}(S) := TC^{ue}(\emptyset) - TC^{ue}(S), \qquad \forall\ S \subseteq N. \tag{6}$$

[5] Its existence is guaranteed, e.g., if the costs of the arcs are continuous functions [9, 37]; in addition, if they are separable and increasing functions of the respective flows, then the uniqueness of the arc equilibrium flow is guaranteed, as discussed below.

[6] This assumption is satisfied for all the case studies reported in this work.

[7] If multiple Wardrop equilibria exist but the arc equilibrium flow is not unique for a given coalition $S \in \mathcal{F}$, different such Wardrop equilibria $\bar{x}(S)$ would be expected to be associated, in general, with distinct total travel times $TC^{ue}(\bar{x}(S))$. In this case, one could redefine $TC^{ue}(S)$ as either the supremum or the infimum of such total travel times (depending on the desired degree of conservativeness in its definition).

(b) When the empty coalition $\emptyset \notin \mathcal{F}$ (i.e., some coalitions are in $\mathcal{F}$ and others do not), one sets

$$v^{ue}(S) := \begin{cases} \left[\max_{S' \in \mathcal{F}_m} TC^{ue}(S')\right] - TC^{ue}(S) & \text{if } S \in \mathcal{F}, \\ 0 & \text{if } S \notin \mathcal{F}, \end{cases} \qquad \forall\ S \subseteq N, \tag{7}$$

where $\mathcal{F}_m \subseteq \mathcal{F}$ is the subset of *minimally connected* coalitions, i.e., the coalitions for which removing any player makes the resulting coalition not an element of $\mathcal{F}$.

An example of computation of the characteristic function v^{ue} is reported in Tables 1–4 of [34, Appendix]; it refers to the same case study considered in Section 5.1. The Wardrop equilibria are computed therein in closed form for each coalition and each value of a scalar traffic demand d, then their associated total costs are evaluated, and finally the characteristic function is computed, based on Definition 3.1.

The following is an interpretation of the characteristic function v^{ue}. In case (a), the utility of the coalition S can be interpreted as its cost-saving with respect to the empty coalition. In case (b), when it is possible to serve the whole demand, the utility of the coalition S can be interpreted as its cost-saving with respect to the maximum cost of a minimally connected coalition; otherwise (i.e., when it is not possible to serve the whole demand), it equals 0. In the first case, subtracting the costs associated with the various coalitions from the maximum cost is made with the aim of mapping disutilities (costs) to utilities (savings).

It was proven in [34] that, in general, the game having the characteristic function v^{ue} reported in one of the two parts of Definition 3.1 is not monotonic, not cohesive, not subadditive, and not convex.

4 Machine learning for Shapley value approximation

4.1 Computational issues in TNc games

Evaluating the Shapley values in TNc games defined on extensive networks presents several computational issues.

The main problem has to do with the required computational effort, which typically increases exponentially with the number of players [24] (for some special cases in which this does not happen, see, e.g., [19, 24]). Only in some contexts, such a computational effort can be reduced by restricting suitably the set of players to some nodes or arcs of interest (see [15, 16]). The computational requirement is even larger for a game whose characteristic function measures traffic congestion [34, 35]. Indeed, as shown in Section 3, in this case, the characteristic function has to be computed by solving a variational inequality for every coalition (see Section 3). A further increase of the computational effort occurs in the presence of parameters (e.g., the traffic demand), since the Shapley values have to be computed for several choices of each parameter.

A *Monte Carlo* approach can be exploited for the approximate evaluation of the Shapley value in TNc games. Loosely speaking, in such an approach, the expectation in the definition of the Shapley value is replaced by an empirical average. This approach is followed, e.g., in [6], where the *ApproShapley algorithm* was proposed for generic TU games. The algorithm was then specialized to TNc games in [15]. In this context, theoretical guarantees such as unbiasedness and bounds on the variance of the Monte Carlo estimate of each Shapley value were obtained in [15]. Recently, a method based on *weighted least-squares regression* (the *Kernel Shap* [22]) was proposed for the approximate evaluation of Shapley values in TU games. It is worth remarking that, although both the Monte Carlo approach and the Kernel Shap method can be exploited to approximate the Shapley values in TNc games whose characteristic functions depend on some parameters, their direct application is limited to fixed choices of such parameters (i.e., they cannot be used directly to generalize their estimates to other values of the parameters, unless they are combined with ML techniques).

4.2 Smooth Shapley values

For certain TNc games, smoothness properties of the Shapley value can be proved. We are interested, in particular, in cases where the Shapley value is a *Lipschitz continuous function*, with Lipschitz constant L, of a parameter.

We shall exploit the following result from [21], which implies Lipschitz continuity of the Shapley value with respect to the traffic demand in the TNc games modeling traffic congestion based on the user equilibrium, defined in Section 3.

Theorem 4.1. *[21] If the arc cost function f_i of every arc i is increasing and continuously differentiable, then the arc equilibrium flow is a Lipschitz continuous function of the traffic demand.*

Corollary 4.2. *Under the assumptions of Theorem 4.1, the Shapley value of each player in a TNc game with the characteristic function v^{ue} is a Lipschitz continuous function of the traffic demand.*

Proof. The statement follows by combining the definition of the Shapley value provided in Equation (1) with one of the two expressions (6) and (7) of the characteristic function v^{ue} provided in Definition 3.1, the expression of the total cost provided in Equation (5), and the statement of Theorem 4.1. □

4.3 Accuracy of Shapley value computation via machine learning

In the following, we discuss an alternative approximate evaluation of the Shapley value, which is based on ML. It can be motivated theoretically, e.g., when the Shapley value is a smooth function of a parameter. This approximation can also be combined with the Monte Carlo approach or the Kernel Shap method, when the training set used to train the learning machine (i.e., to find the optimal values of its parameters) is noisy. In the univariate case, such an approximation is based on the following theoretical result.

Proposition 4.3. *Let I be a bounded and closed interval with length $\mu(I)$, $f : I \to \mathbb{R}$ be Lipschitz continuous with Lipschitz constant L, $\sigma : \mathbb{R} \to \mathbb{R}$ be a sigmoidal activation function[8], and $M \in \mathbb{N} \setminus \{0\}$. Then,*

$$\inf_{\substack{a_m, b_m, c_m \in \mathbb{R}, \\ \text{with } m=1,\ldots,M}} \sup_{z \in I} \left| f(z) - \sum_{m=1}^{M} c_m \sigma(a_m z + b_m) \right| \leq \frac{c_\sigma \mu(I) L}{M+1}, \tag{8}$$

where $c_\sigma > 0$ depends only on the choice of σ[9].

Proof. The result follows by an application of [11, Theorem 1] on the nonlinear approximation of univariate functions of bounded variation by one-hidden-layer feedforward artificial neural networks, as a Lipschitz continuous function f with Lipschitz constant L is also of bounded variation on a bounded and closed interval I, with its variation $V(f)$ bounded from above by $\mu(I)L$. □

According to Proposition 4.3, under the Lipschitz continuity property (with constant L), in the univariate case (i.e., when $|W| = 1$ and $d_w = d$), one can approximate the Shapley value of every arc on any bounded and closed interval I by using a one-hidden-layer feedforward artificial neural network with a sigmoidal activation function σ and M computational units, i.e., one having the form

$$f_M(d) = \sum_{m=1}^{M} c_m \sigma(a_m d + b_m), \tag{9}$$

[8] I.e., a bounded, differentiable, and real-valued function defined for all real input values, having non-negative derivative at each point, and exactly one inflection point.

[9] To get a simpler expression, the right-hand side of (8) can also be bounded from above by $(c_\sigma \mu(I) L)/M$.

obtaining a sup-norm approximation error on I smaller than or equal to $\frac{c_\sigma \mu(I) L}{M+1} + \varepsilon$ (with $\varepsilon > 0$ arbitrarily small), where $a_m, b_m, c_m \in \mathbb{R}$, with $m = 1, \ldots, M$, are parameters to be optimized during the training of the network.

Moreover, Proposition 4.3 suggests the application of supervised ML methods to approximate each Shapley value function[10] (say, $f(d)$) by using a neural network of the form (9) and a training set made of a finite number of pairs of the form $(d_h, f(d_h))$, for $h = 1, \ldots, H \in \mathbb{N} \setminus \{0\}$.

It is worth observing that higher-order smoothness of the Shapley value with respect to the vector of demands could be needed to extend the upper bound on the sup-norm approximation error of Proposition 4.3 to the case in which the Shapley value is a function of two or more parameters (although Proposition 4.3 could be still applied directly if its domain were restricted to a segment, because a univariate function would be obtained in this way). However, smoothness higher than Lipschitz continuity is not expected to hold in general, because a change in the vector of demands could also change the set of positive path flows $\bar{x}_p(S)$ at the Wardrop equilibrium, possibly resulting in loss of differentiability of the Shapley value[11]. Such a loss of differentiability could be also motivated by the fact that, in several cases, solving the variational inequality (4) is equivalent to solving a suitable constrained optimization problem[12] [25], for which a change in the vector of demands could also change the set of active constraints at optimality.

5 Case study 1: the Braess' network

Braess' paradox [3, 4] highlights the reason why adding one resource to a network may, in some cases, deteriorate, instead of improving, the overall network performance. Indeed, the Wardrop equilibrium can be interpreted as a Nash equilibrium with an infinite number of non-atomic players (i.e., having a negligible individual impact on the outcome of the game), and Nash equilibria are typically not efficient from a social planner perspective (see also the concepts of *price of anarchy* and *price of stability* [41]). For any positive integer n, a special graph with n vertices (which is known in the literature as n-th *Braess' graph* [40]) exists, for which it is possible to prove that such a phenomenon occurs. Although originally (and still typically) studied in the framework of transportation networks, Braess' paradox is also relevant in several other contexts of application, such as in the cases of telecommunications networks and the Internet, mechanical and electrical networks, metabolic networks, and even sports analytics (see [27] for a recent review article on the Braess' paradox).

In this section, we consider the TNc game (N, v^{ue}) defined on the network represented in Figure 1 and introduced by Dietrich Braess in [3], in which the paradox was first discovered. The unique Origin-Destination pair is $(1, 4)$ and we assume that the set of all the players coincides with the set of all the arcs, i.e., $N = \{q, r, s, t, u\}$. Accordingly, the characteristic function v^{ue} is expressed by part (b) of Definition 3.1. In Section 5.1 we will analyze the approximation of the Shapley value of every arc as a function of the traffic demand, while in Section 5.2 the approximation of the Shapley values as functions of both the traffic demand and an arc cost parameter will be investigated.

[10] This remark extends to the case of the approximation of the percentage Shapley value functions considered later in Equation (11). In fact, when the Shapley value functions are Lipschitz continuous, also the percentage Shapley value functions are Lipschitz continuous, provided that their denominators are bounded away from 0. In this case, indeed, they are compositions of Lipschitz continuous functions, hence they are Lipschitz continuous, too.

[11] This loss of differentiability is observed, e.g., in the case study related to the Braess' network, which is reported in Section 5.

[12] Hence, in these cases, an alternative way to find $\bar{x}(S)$ and the corresponding $\lambda(S)$ consists first in solving that optimization problem (finding $\bar{x}(S)$), then determining $\lambda(S)$ directly from the equilibrium conditions (3).

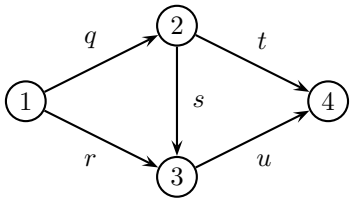


Figure 1: Braess' network.

5.1 Shapley values as functions of the traffic demand

The arc cost functions are defined as follows (see [3]):

$$\begin{aligned} c_q(f) &= 10 f_q, \\ c_r(f) &= f_r + 50, \\ c_s(f) &= f_s + 10, \\ c_t(f) &= f_t + 50, \\ c_u(f) &= 10 f_u, \end{aligned} \tag{10}$$

where $f_q, \ldots, f_u$ denote the arc flows, and we assume the traffic demand d from node 1 to node 4 varies in the interval $[1, 10]$. To better compare the arcs in terms of the Shapley value, we computed the *percentage Shapley value* of every arc i, defined as[13]

$$PercSh(i) = \frac{Sh(i)}{\sum\limits_{j \in N} Sh(j)} \cdot 100\%. \tag{11}$$

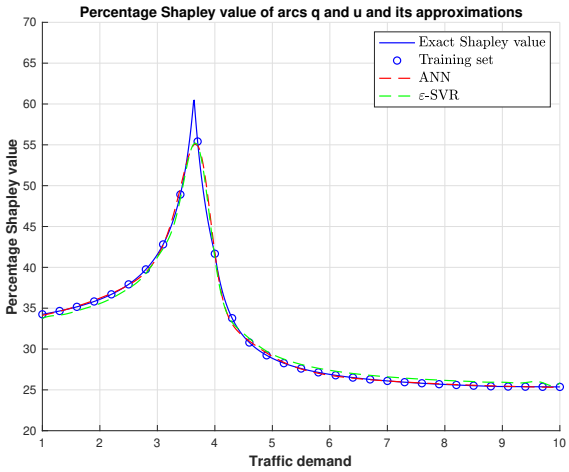
We tested two ML models to approximate the percentage Shapley value of every arc: a *one-hidden-layer feedforward artificial neural network* of the form (9) and an *ε-support vector regression* [44].

The neural network had $M = 20$ computational units, the training was based on the minimization of the *square loss function*, while the training inputs were $d_h = 1 + 0.3(h-1)$ with $h = 1, \ldots, H = 31$. The neural network training was performed by *batch gradient descent* implemented by *backpropagation*, with a *learning rate* equal to 0.01 and $10,000$ *training epochs*. For scaling reasons, the input d was mapped linearly onto the interval $[0, 1]$ and the weights a_m, b_m, c_m of all the computational units in (9) were initialized by realizations of random variables, uniformly and independently distributed on the interval $[-1, 1]$.

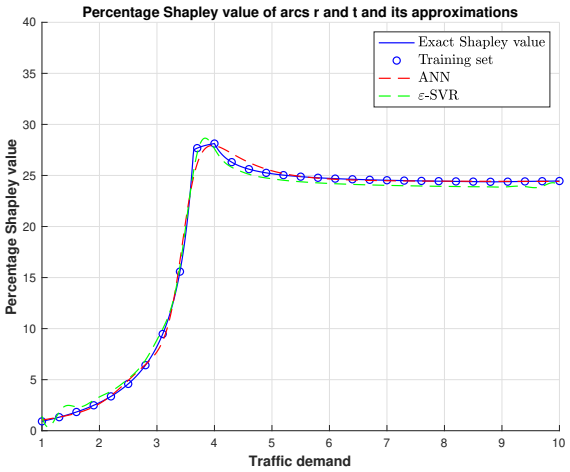
In the ε-support vector regression model we used the same training set as in the neural network model, we set the width of the *ε-tube* to $\varepsilon = 0.5$, the *soft margin parameter* to $C = 100$ and a *Gaussian kernel* with parameter $\gamma = 10$ was chosen.

The values predicted by both models were then plotted on the interval $[1, 10]$ using a finer discretization with a step size of 0.01. Figure 2 shows, for each arc of the network, the exact percentage Shapley value (blue line), which can be computed in closed form[14] for this case study [34], the training points (blue circles) and the percentage Shapley value predicted by the *Artificial Neural Network* model (*ANN*, red dashed line) and the *ε-Support Vector Regression* model (*ε-SVR*, green dashed line). In both cases, the resulting approximations look quite good, apart from the region around the spikes in parts (A) and (C) of Figure 2, which are likely due to the small density of training points in that region[15].

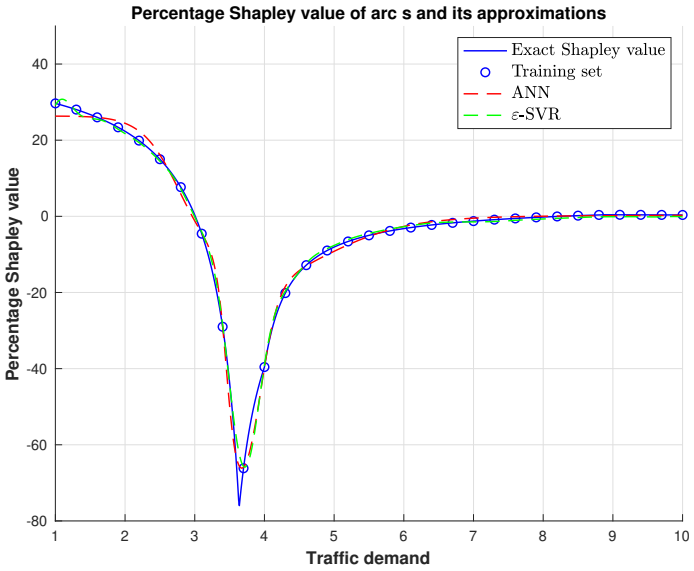
[13] It is worth observing that, due to the efficiency property of the Shapley value, the denominator $\sum\limits_{j \in N} Sh(j)$ of each percentage Shapley value is the total utility of the grand coalition N. This is bounded away from 0 for all the case studies considered in



(a) Arcs q and u.



(b) Arcs r and t.



(c) Arc s.

Figure 2: Braess' network: percentage Shapley value of every arc as a function of the traffic demand: exact values and predicted values by the ANN and ε-SVR models.

Table 1: Braess' network with varying traffic demand: average absolute error between the exact percentage Shapley value of every arc and the percentage Shapley value predicted by the ANN and ε-SVR models; average normalized Kendall distance between the ranking of players based on the exact Shapley values and their ranking based on the Shapley values predicted by either the ANN or ε-SVR model.

	Arc	ANN	ε-SVR
Avg. absolute error	q	0.2462	0.5800
	r	0.2064	0.4997
	s	1.0795	0.8040
	t	0.2064	0.4997
	u	0.2462	0.5800
Avg. norm. Kendall distance		0.0866	0.0007

Table 1 shows the *average absolute error*, computed on the finer discretization, between the exact percentage Shapley value of every arc and the percentage Shapley value predicted by the ANN and ε-SVR models. We note that almost all values are less than 1 and that the ANN model provided better predictions than ε-SVR for all the arcs except for the arc s. Moreover, in the last row of Table 1 we show the *average normalized Kendall τ distance* between the ranking of players based on the exact percentage Shapley values and their ranking based on the percentage Shapley values predicted by either the ANN or ε-SVR model. We recall that, given two permutations $\pi = (\pi_1, \ldots, \pi_n)$ and $\varphi = (\varphi_1, \ldots, \varphi_n)$ of the numbers $1, \ldots, n$, the *normalized Kendall τ ranking distance* [18] between π and φ is defined as

$$K(\pi, \varphi) = \frac{|\{(i,j) : \ i < j, \ \ (\pi_j - \pi_i)(\varphi_j - \varphi_i) < 0\}|}{n(n-1)/2}.$$

Notice that the ε-SVR model provided better performance than the ANN model in terms of the ranking of players. This is not in contrast with the results reported above regarding the average absolute error, since this is a different metric.

5.2 Shapley values as functions of the traffic demand and arc cost

In this section, we assume that the traffic demand d from node 1 to node 4 varies in the interval $[1, 10]$ and that the arc cost functions are defined as follows:

$$\begin{aligned} c_q(f) &= 10 f_q, \\ c_r(f) &= f_r + 50, \\ c_s(f) &= f_s + a, \\ c_t(f) &= f_t + 50, \\ c_u(f) &= 10 f_u, \end{aligned} \tag{12}$$

the work.

[14] Its closed-form expression can also be obtained from the next Equations (13)–(15), specialized to the case $a = 10$.

[15] It is worth remarking that some players have the same Shapley value for all values of d (see the next footnote 16 for a justification of this behavior). By exploiting this *a priori* knowledge, a single ML model is trained for each of such players. Indeed, no changes would be obtained by training a separate ML model for each such player (all other things being equal, e.g., the initialization of the weights in the case of the ANN model), since the two respective training sets are essentially the same. In the case of noisy labels, however, training a single ML model for two players with the same Shapley value (e.g., by joining the two training sets associated with each such player) is expected to be preferable to training a separate ML model for each of them. Indeed, in that case, one would get two different noisy estimates of the same label (i.e., the common Shapley value) for each value of the demand d in the training set, instead of one.

where the parameter a varies in the interval $[0, 20]$. Since the user equilibrium can be computed in closed form for any value of the pair of parameters (d, a), after some computations it is possible to get the following explicit formulas for the exact Shapley value of every arc[16] as a function of the pair of parameters (d, a):

$$Sh(q) = Sh(u) = \begin{cases} \dfrac{d\,(2000 - 367d - 40a)}{120} & \text{if } d \leq \dfrac{50-a}{11}, \\ \dfrac{d\,(2610d + 151a - 7550)}{1560} & \text{if } d \in \left[\dfrac{50-a}{11}, \dfrac{50-a}{10}\right], \\ \dfrac{d\,(7030d + 593a - 29650)}{1560} & \text{if } d \in \left[\dfrac{50-a}{10}, \dfrac{50-a}{4.5}\right], \\ \dfrac{d\,(466d + 29a - 1450)}{120} & \text{if } d \geq \dfrac{50-a}{4.5}, \end{cases} \tag{13}$$

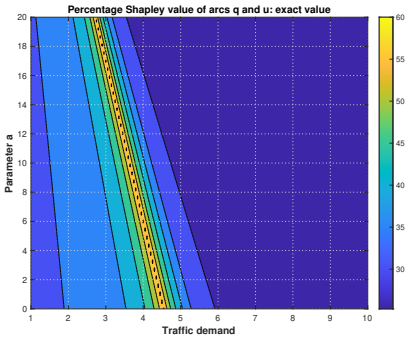
$$Sh(r) = Sh(t) = \begin{cases} \dfrac{11\,d^2}{40} & \text{if } d \leq \dfrac{50-a}{11}, \\ \dfrac{11d\,(235d + 19a - 950)}{1040} & \text{if } d \in \left[\dfrac{50-a}{11}, \dfrac{50-a}{10}\right], \\ \dfrac{3d\,(1555d + 139a - 6950)}{1040} & \text{if } d \in \left[\dfrac{50-a}{10}, \dfrac{50-a}{4.5}\right], \\ \dfrac{3d\,(103d + 7a - 350)}{80} & \text{if } d \geq \dfrac{50-a}{4.5}, \end{cases} \tag{14}$$

$$Sh(s) = \begin{cases} \dfrac{d\,(500 - 133d - 10a)}{30} & \text{if } d \leq \dfrac{50-a}{11}, \\ \dfrac{d\,(465d + 151a - 7550)}{1560} & \text{if } d \in \left[\dfrac{50-a}{11}, \dfrac{50-a}{10}\right], \\ \dfrac{d\,(985d + 203a - 10150)}{1560} & \text{if } d \in \left[\dfrac{50-a}{10}, \dfrac{50-a}{4.5}\right], \\ \dfrac{d\,(d - a + 50)}{120} & \text{if } d \geq \dfrac{50-a}{4.5}. \end{cases} \tag{15}$$

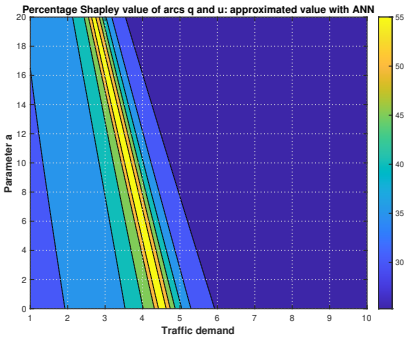
Also in this case we tested the ANN and ε-SVR models to approximate the percentage Shapley value of every arc. Training inputs $(d_h, a_{h'})$ were chosen as $d_h = 1 + 0.2(h-1)$, with $h = 1, \ldots, 46$, and $a_{h'} = 0.4(h'-1)$, with $h' = 1, \ldots, 51$. The parameters of the ANN model were the same as in Section 5.1, while we set parameters $\varepsilon = 0.0001$, $C = 500$ and $\gamma = 500$ for the ε-SVR model.

The predicted values by both models were then plotted on the box $[1, 10] \times [0, 20]$ using a finer discretization with step sizes of 0.05 and 0.1, respectively. Figure 3 shows the level sets of the percentage Shapley value as a function of the pair of parameters (d, a). The exact value for each arc is displayed in subfigures A, D, G, while the predicted values by the ANN and ε-SVR models are in subfigures B, E, H and C, F, I, respectively.

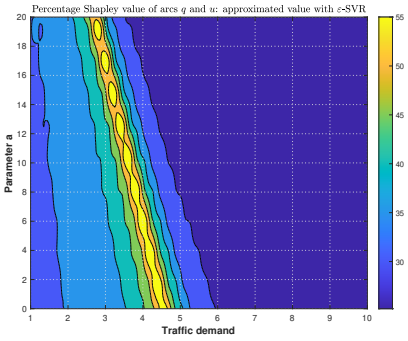
[16] Interestingly, some pairs of players (arcs q and u in one case, arcs r and t in another case) turn out to have, in general (i.e., for all values of d), the same Shapley value. This occurs despite the fact that, in general, they are not symmetric players in the sense of the theory of TU games. For instance, it turns out from [34, Tables 1–4] that, in general, $v^{ue}(\{q, t\}) \neq v^{ue}(\{t, u\})$ or $v^{ue}(\{q, r, s\}) \neq v^{ue}(\{r, s, u\})$, which proves that, in general, q and u are not symmetric players. Nevertheless, since it still follows from [34, Tables 1–4], due to the identical functional form of some arc cost functions in (12), that other coalitions have, in general, the same value of the characteristic function (e.g., $v^{ue}(\{q, t\}) = v^{ue}(\{r, u\})$, and $v^{ue}(\{r, s, u\}) = v^{ue}(\{q, s, t\})$), it turns out that q and u have, in general, the same Shapley value.



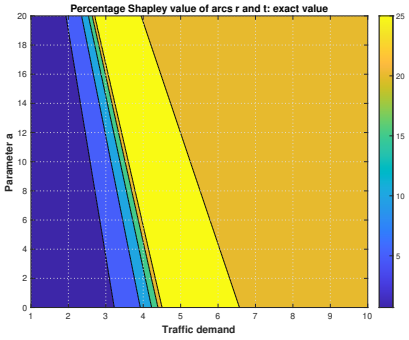
(a) Exact values for the arcs q and u.



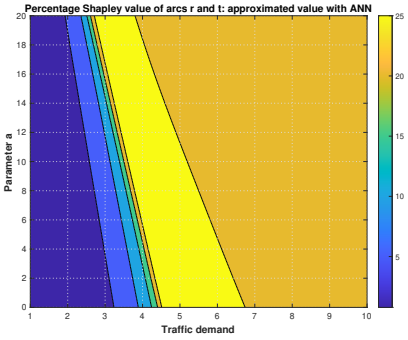
(b) Approximated values for the arcs q and u with ANN.



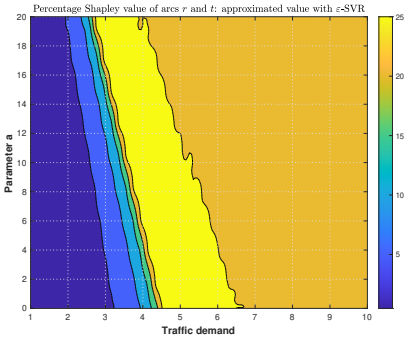
(c) Approximated values for the arcs q and u with ε-SVR.



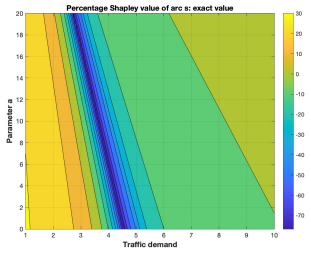
(d) Exact values for the arcs r and t.



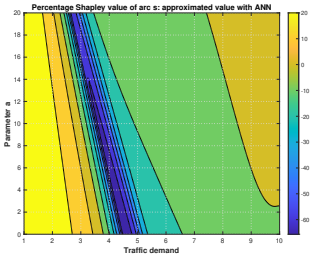
(e) Approximated values for the arcs r and t with ANN.



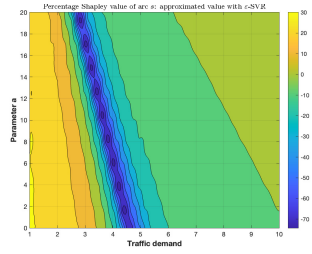
(f) Approximated values for the arcs r and t with ε-SVR.



(g) Exact value for the arc s.



(h) Approximated value for the arc s with ANN.



(i) Approximated value for the arc s with ε-SVR.

Figure 3: Braess' network: percentage Shapley value of every arc as a function of the traffic demand d and the parameter a: exact values and predicted values by ANN and ε-SVR models.

Table 2: Braess' network with two varying parameters: average absolute error between the exact percentage Shapley value of every arc and the percentage Shapley value predicted by the ANN and ε-SVR models; average normalized Kendall distance between the ranking of players based on the exact Shapley values and their ranking based on the Shapley values predicted by either the ANN or ε-SVR model.

	arc	ANN	ε-SVR
Avg. absolute error	q	0.1220	0.1327
	r	0.0777	0.0656
	s	0.8442	0.3821
	t	0.0777	0.0656
	u	0.1220	0.1327
Avg. norm. Kendall distance		0.0967	0.0003

Table 2 shows the average absolute error between the exact percentage Shapley value of every arc and the percentage Shapley value predicted by the ANN and ε-SVR models. Moreover, the last row shows the average normalized Kendall distance between the ranking of players based on the exact Shapley values and their ranking based on the Shapley values predicted by either the ANN or ε-SVR model. Notice that, also in this case, the ε-SVR model provided a very good approximation of the exact ranking of players when averaging with respect to all the values considered for the pair of parameters (d, a).

6 Case study 2: the Sioux-Falls network

In order to investigate a medium-sized network, having both a realistic topology and realistic demands, in this section we consider the well-known *Sioux-Falls network* with 24 nodes and 76 arcs, which is used quite often in the literature as a benchmark for network analysis and design. Its topology is illustrated in Figure 4.

The arc cost functions are of the *Bureau of Public Roads (BPR)* form [5]:

$$c_i(f_i) = t_i^0 \left[1 + 0.15 \left(\frac{f_i}{u_i} \right)^4 \right],$$

where f_i denotes the flow on arc i, while t_i^0 and u_i represent the free flow travel time and the capacity of arc i, respectively. The values of the parameters t_i^0, u_i and the traffic demands of the 528 origin-destination pairs are given in [33].

We consider the TNc game (N, v^{ue}) where the set N of critical arcs consists of 10 arcs with the highest capacity values, i.e., $N = \{1, 3, 7, 18, 35, 37, 38, 54, 56, 60\}$. Accordingly, the characteristic function v^{ue} is expressed by part (a) of Definition 3.1. To analyze the percentage Shapley values of these arcs as functions of the traffic demand, the original traffic demands were multiplied by a factor $\delta \in (0, 1]$. The user equilibrium for each value of δ was calculated using the path-based algorithm proposed in [32].

We tested the ANN and ε-SVR models to estimate the percentage Shapley value of every arc by using three different training sets: a training set with exact values, a training set with approximated values obtained by exploiting the Monte Carlo method [6] and a training set with approximated values obtained by exploiting the Kernel Shap method [22]. All the training sets were generated with inputs $\delta_h = 0.1 + 0.05(h - 1)$, with $h = 1, \ldots, 19$. In the case of the Monte Carlo method, the approximated percentage Shapley values were computed, for each value of δ_h, by using one sample with 30 random permutations of the set of players.

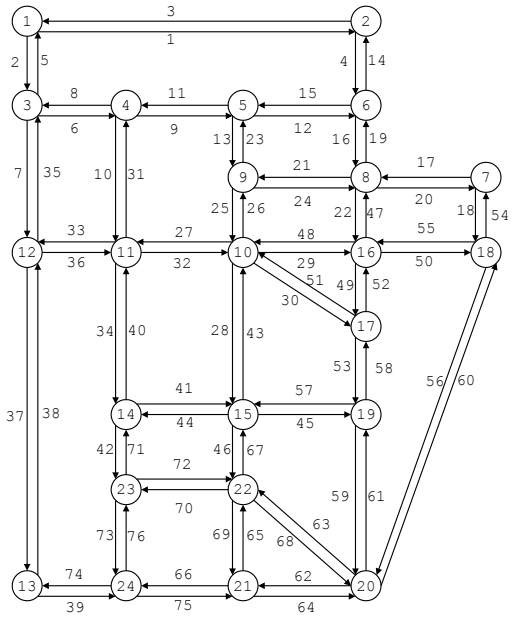
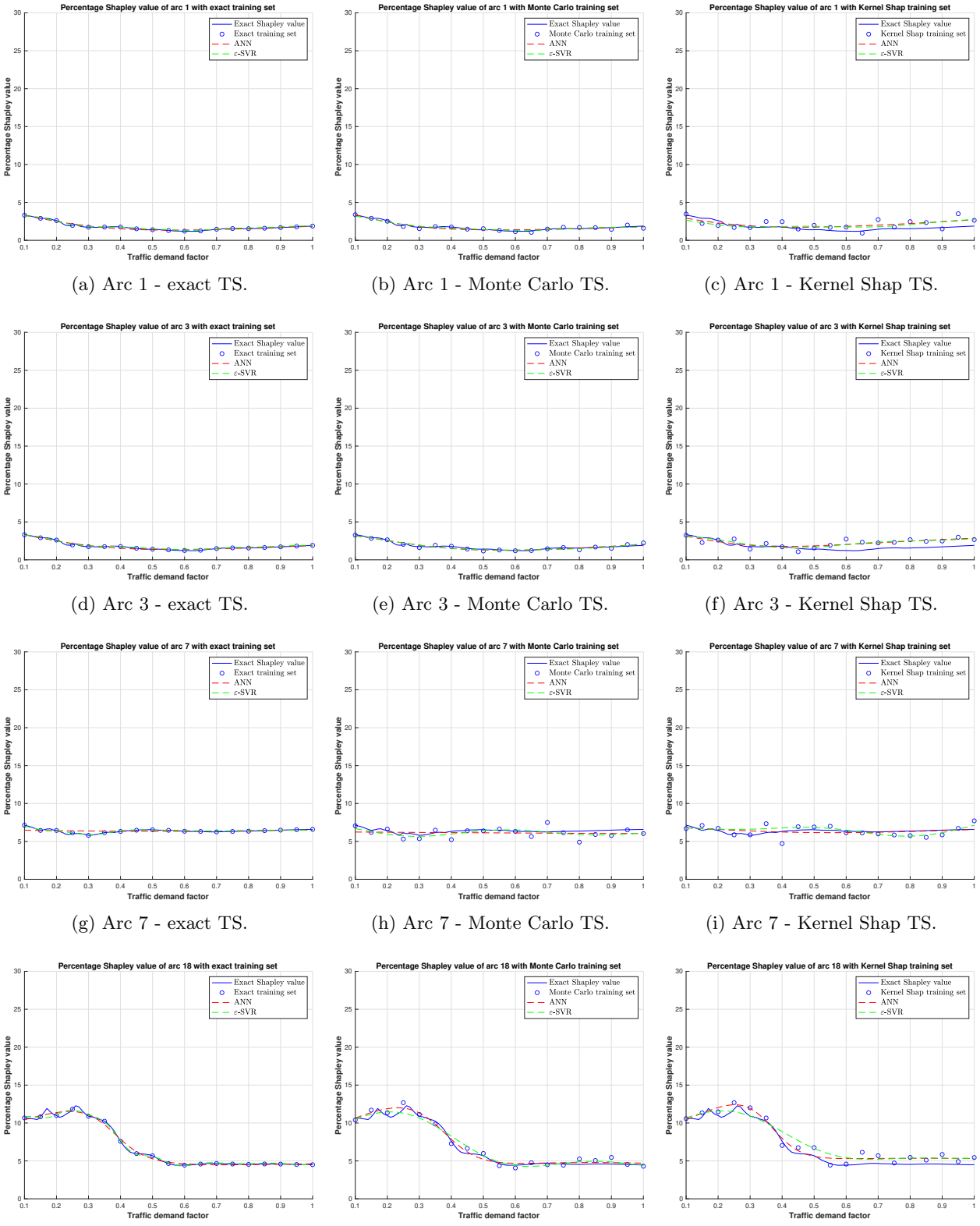


Figure 4: Sioux-Falls network.

In the case of the Kernel Shap method, the approximated percentage Shapley values were computed by exploiting the coalitions generated in the application of the Monte Carlo method.

The values predicted by both models were then plotted on the interval $[0.1, 1]$ using a finer discretization with a step size of 0.01. Figures 5-7 show the percentage Shapley value of each player predicted by the ANN and ε-SVR models for each of the three considered training sets. In this case, the performance obtained by the ε-SVR model was generally better than the one of the ANN model for the training set with exact values, whereas the ANN model outperformed the ε-SVR model for the other two training sets containing approximated values.

Table 3 shows: (i) the average absolute error between the exact percentage Shapley value of every arc and the percentage Shapley value predicted by the ANN and ε-SVR models, (ii) the average normalized Kendall distance between the ranking of players based on the exact Shapley values and their ranking based on the Shapley values predicted by either the ANN or ε-SVR model for each of the considered training sets, (iii) the computational time of each solution approach. Notice that, in the case of the training set with the exact values, the ε-SVR model provided a very good approximation of the exact ranking of players when averaging with respect to all the values considered for the parameter δ_h. For the other two training sets, the ANN and ε-SVR models achieved a similar performance (although, as expected, a bit worse with respect to the case of the training set with exact values). We note that the approaches based on the approximated training sets provided on the one hand good results both in terms of the average absolute error and in terms of the average normalized Kendall distance, and on the other hand they allowed a drastic reduction in the computational time. In fact, computing the exact percentage Shapley values in the range $[0.1, 1]$ with a discretization step of 0.01 took about 12.68 hours while calculating the values predicted by the models with approximated training sets took only 0.65 hours, approximately 20 times less.



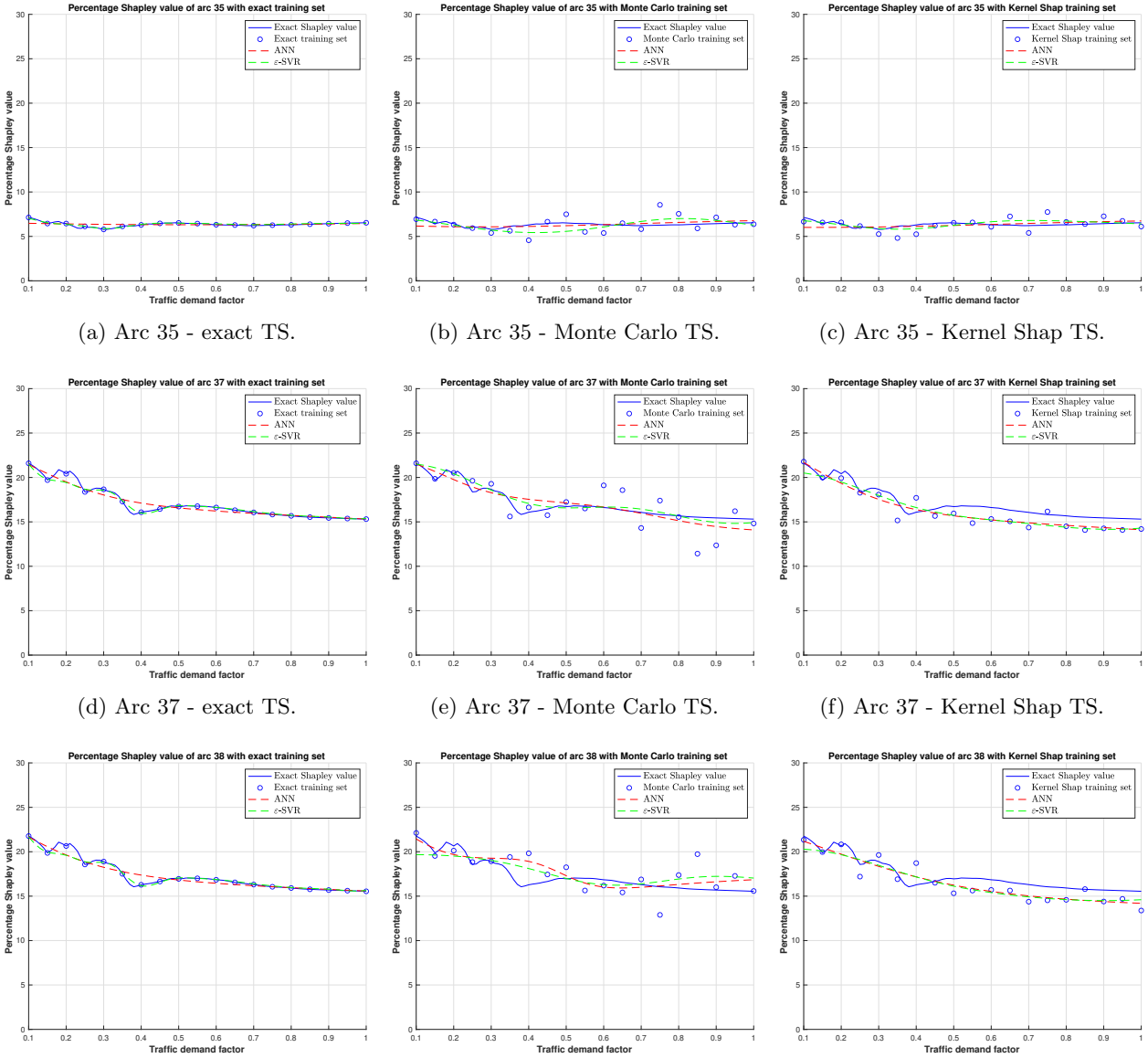
(a) Arc 1 - exact TS. (b) Arc 1 - Monte Carlo TS. (c) Arc 1 - Kernel Shap TS.

(d) Arc 3 - exact TS. (e) Arc 3 - Monte Carlo TS. (f) Arc 3 - Kernel Shap TS.

(g) Arc 7 - exact TS. (h) Arc 7 - Monte Carlo TS. (i) Arc 7 - Kernel Shap TS.

(j) Arc 18 - exact TS. (k) Arc 18 - Monte Carlo TS. (l) Arc 18 - Kernel Shap TS.

Figure 5: Sioux-Falls network: percentage Shapley values of arcs 1, 3, 7, 18 as functions of the traffic demand: exact values and predicted values by the ANN and ε-SVR models with an exact, Monte Carlo, and Kernel Shap Training Set (TS).

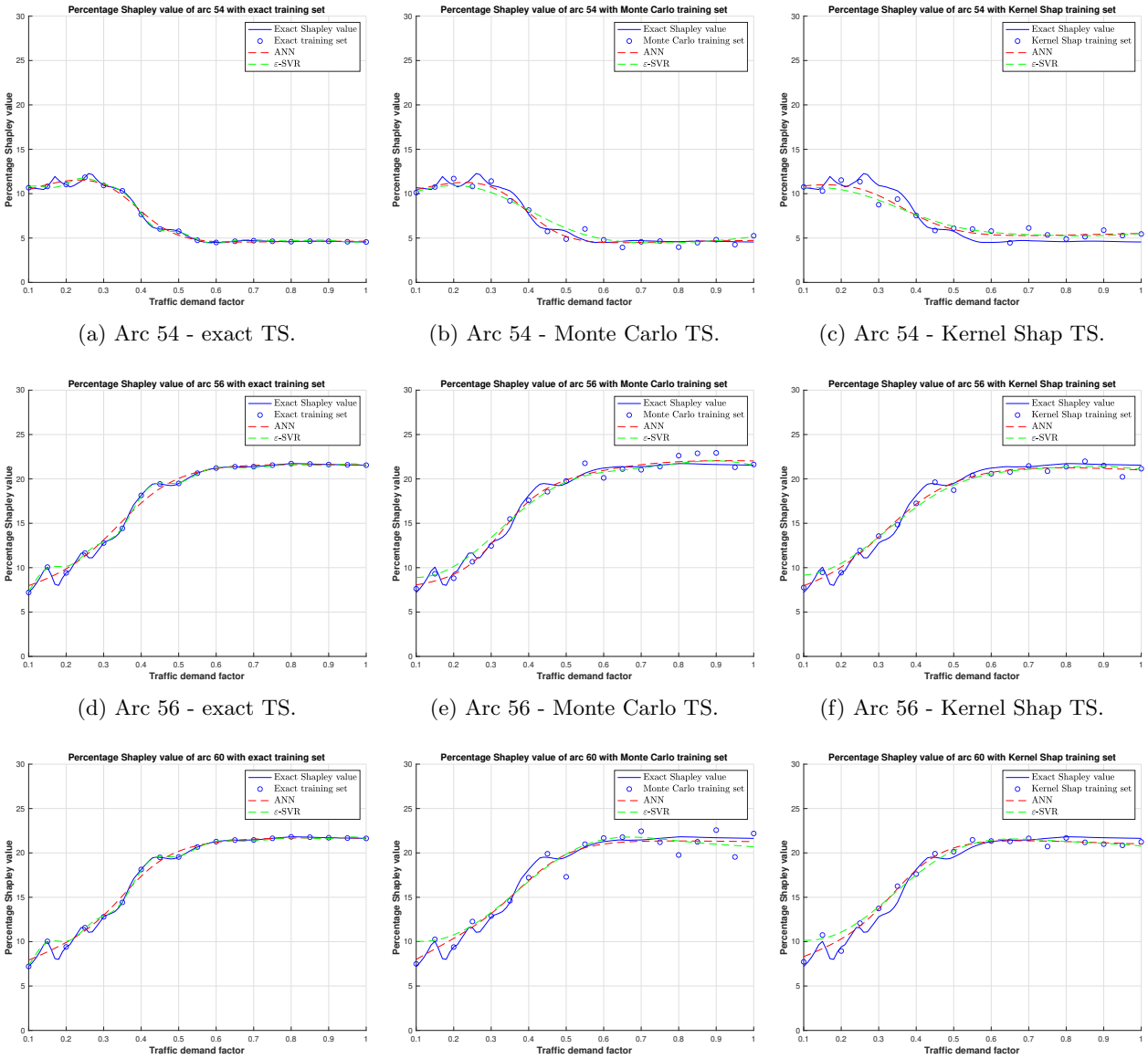


(a) Arc 35 - exact TS. (b) Arc 35 - Monte Carlo TS. (c) Arc 35 - Kernel Shap TS.

(d) Arc 37 - exact TS. (e) Arc 37 - Monte Carlo TS. (f) Arc 37 - Kernel Shap TS.

(g) Arc 38 - exact TS. (h) Arc 38 - Monte Carlo TS. (i) Arc 38 - Kernel Shap TS.

Figure 6: Sioux-Falls network: percentage Shapley values of arcs 35, 37, 38 as functions of the traffic demand: exact values and predicted values by the ANN and ε-SVR models with an exact, Monte Carlo, and Kernel Shap Training Set (TS).



(a) Arc 54 - exact TS. (b) Arc 54 - Monte Carlo TS. (c) Arc 54 - Kernel Shap TS.

(d) Arc 56 - exact TS. (e) Arc 56 - Monte Carlo TS. (f) Arc 56 - Kernel Shap TS.

(g) Arc 60 - exact TS. (h) Arc 60 - Monte Carlo TS. (i) Arc 60 - Kernel Shap TS.

Figure 7: Sioux-Falls network: percentage Shapley values of arcs 54, 56, 60 as functions of the traffic demand: exact values and predicted values by the ANN and ε-SVR models with an exact, Monte Carlo, and Kernel Shap Training Set (TS).

Table 3: Sioux-Falls network: average absolute error between the exact percentage Shapley value of every arc and the percentage Shapley value predicted by the ANN and ε-SVR models with exact or approximated training sets; average normalized Kendall distance between the ranking of players based on the exact Shapley values and their ranking based on the Shapley values predicted by either the ANN or ε-SVR model.

		Exact TS		Monte Carlo TS		Kernel Shap TS	
	arc	ANN	ε-SVR	ANN	ε-SVR	ANN	ε-SVR
Avg. absolute error	1	0.0817	0.0817	0.0908	0.0911	0.4513	0.4487
	3	0.0792	0.0818	0.1098	0.1134	0.5454	0.5188
	7	0.1423	0.0701	0.3082	0.3089	0.2207	0.3738
	18	0.1862	0.1567	0.2759	0.3380	0.6204	0.7513
	35	0.1426	0.0694	0.2321	0.4396	0.2391	0.2650
	37	0.3389	0.1936	0.5446	0.3740	0.9496	0.9675
	38	0.3427	0.1980	0.8164	0.8922	0.9460	0.9852
	54	0.1846	0.1548	0.2658	0.4499	0.7408	0.8992
	56	0.3179	0.2247	0.4020	0.4727	0.5337	0.6358
	60	0.3249	0.2219	0.4903	0.7037	0.5631	0.7538
Avg. norm. Kendall distance		0.0208	0.0129	0.0796	0.0752	0.0440	0.0559
Time (h)		2.76	2.76	0.65	0.65	0.65	0.65

Table 4: Sioux-Falls network: the average absolute error between the exact percentage Shapley value of every arc and the percentage Shapley value predicted by the two ML models (ANN and ε-SVR) and the two approximation methods (Monte Carlo and Kernel Shap); the average normalized Kendall distance between the ranking of players based on the exact Shapley values and the ranking based on the Shapley values predicted by the two ML models and the two approximation methods.

	arc	ANN with MC	ε-SVR with MC	MC on test set	ANN with KS	ε-SVR with KS	KS on test set
Avg. absolute error	1	0.0906	0.0911	0.0926	0.4520	0.4487	0.6607
	3	0.1110	0.1134	0.0853	0.5394	0.5188	0.6416
	7	0.3080	0.3089	0.4628	0.1791	0.3738	0.4955
	18	0.2827	0.3380	0.4661	0.6181	0.7513	0.6492
	35	0.2433	0.4396	0.5158	0.2485	0.2650	0.4768
	37	0.5593	0.3740	1.0123	0.9463	0.9675	1.1063
	38	0.8307	0.8922	1.0350	0.9466	0.9852	1.0932
	54	0.2352	0.4499	0.4906	0.7467	0.8992	0.6854
	56	0.4021	0.4727	0.7409	0.5392	0.6358	0.5301
	60	0.4960	0.7037	0.7924	0.5637	0.7538	0.5851
	avg	0.3559	0.4184	0.5694	0.5780	0.6599	0.6924
Avg. norm. Kendall distance		0.0796	0.0752	0.0632	0.0440	0.0559	0.0689
Time (h)		0.65	0.65	2.28	0.65	0.65	2.28

Finally, we show a performance comparison between the ANN and ε-SVR models and the Monte Carlo and Kernel Shap approximation methods. Specifically, we trained the ANN and ε-SVR models on the Monte Carlo (resp., Kernel Shap) training set generated as in the previous analysis. Then, we compared the percentage Shapley values predicted by the ANN and ε-SVR models on the test set, i.e., the set of points of the interval $[0.1, 1]$ with a step size of 0.01, excluding the points of the training set, with the corresponding Shapley values predicted by the Monte Carlo (resp., Kernel Shap) method applied to the whole test set. Table 4 shows: (i) the average absolute error between the exact percentage Shapley value of every arc and the percentage Shapley value predicted by the ANN and ε-SVR models trained on a Monte Carlo (resp., Kernel Shap) training set and the Monte Carlo (resp., Kernel Shap) method applied to the whole test set; (ii) the average normalized Kendall distance between the ranking of players based on the exact Shapley values and their ranking based on the Shapley values predicted by the two ML models and the Monte Carlo and Kernel Shap methods; (iii) the computational time of each solution approach. The numerical results show that the two ML models outperform the Monte Carlo and the Kernel Shap methods in terms of both accuracy and computational time. Indeed, the Shapley values predicted by the Monte Carlo and Kernel Shap methods applied to the whole test set are almost always worse than those predicted by the ML models, and their computational time is more than 3 times longer than that of the ML models (2.28 versus 0.65 hours). Figure 8 shows the Shapley values predicted by the two ML models and by the Monte Carlo and Kernel Shap methods for the arc 37.

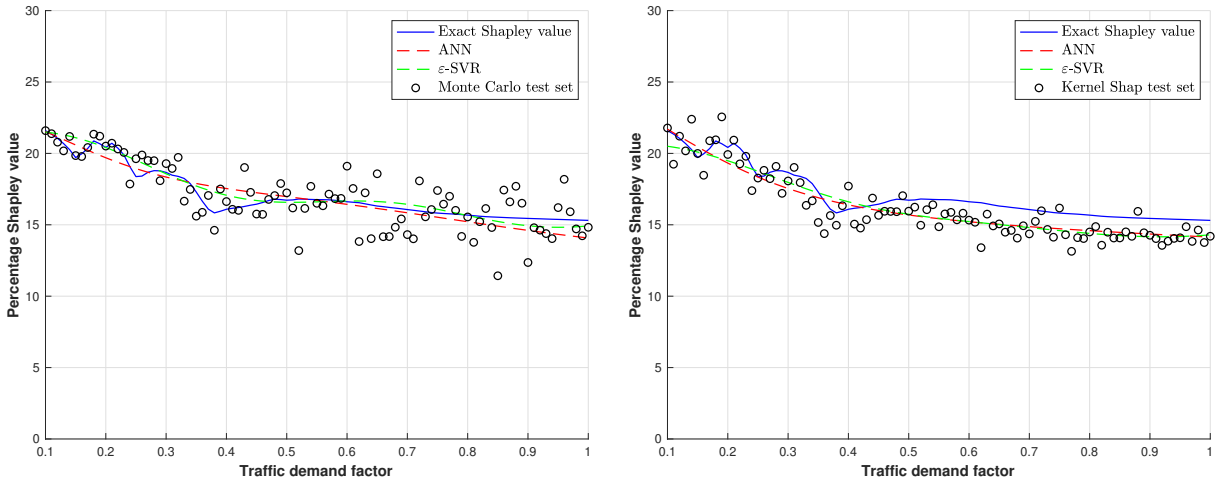


Figure 8: Sioux-Falls network: the percentage Shapley value of arc 37 as a function of the traffic demand. (Left) Comparison between predicted values by the two ML models trained on a Monte Carlo training set, and the predictions achieved by the Monte Carlo method applied to the whole test set. (Right) Comparison between predicted values by the two ML models trained on a Kernel Shap training set, and the predictions achieved by the Kernel Shap method applied to the whole test set.

7 Discussion

In this article, we have examined the Lipschitz continuity of the Shapley value for a class of parameter-dependent transportation network cooperative games. Then, we have exploited such a property to motivate the use of machine learning techniques for the approximate computation of the Shapley value in such games.

Variations of the results (e.g., other conditions under which the Shapley value is a smooth function of some parameters) are expected to hold for other families of transferable-utility cooperative games on graphs: for instance, the cases in which the Wardrop equilibrium considered in Section 3 is replaced either by the Nash equilibrium of a noncooperative game model with a finite number of players, in which each player represents a group of users [20], or by the system optimum (see, e.g., [9]), which involves the optimal behavior of a social planner. Such cases could be analyzed, e.g., using the parametric variational inequality approach studied in [36]. Another possible extension refers to the case in which players represent, e.g., joints of the human body or individuals in a group of people, and the Shapley value is used as a measure of the importance of such players for transferable-utility cooperative game models (parametrized, for instance, by time) that capture the overall motion behavior via suitable movement features [19]. Finally, the results obtained in the work could be extended to the *generalized Shapley value* arising in a *generalized coalitional game* model [30], since it has a similar expression as the Shapley value, and it has also relevant economic applications (e.g., in the case of sports analytics [23, 2]).

Acknowledgments

The authors were partially supported by the PRIN project 2022S8XSMY and the project F/310027/02/X56, funded by the Italian Ministry of Enterprises and Made in Italy. G. Gnecco was partially supported by the GNAMPA-INdAM 2023 project "Development of machine-learning methods for the estimation of the Shapley value and of its generalizations" (code CUP: E53C22001930001), the GNAMPA-INdAM 2024 project "Variational inequality approach to network games", and the project "ROBOFARM" (code CUP:

D63C23000520009), funded by the Tuscany Region. M. Sanguineti was partially supported by the FISA-2022-00827 project "UAV-FIRE" ("UAV platform for monitoring forest fire outbreaks and detecting post-fire recovery hotspots"), and the project of the PDGP DIT.AD021.104 of the Institute of Marine Engineering, National Research Council of Italy, where he is research associate. The authors are members of the Gruppo Nazionale per l'Analisi Matematica, la Probabilità e le loro Applicazioni (GNAMPA - National Group for Mathematical Analysis, Probability and their Applications) of the Instituto Nazionale di Alta Matematica (INdAM - National Institute of Higher Mathematics). M. Sanguineti is Guest Scholar at IMT – School for Advanced Studies, Lucca.

Dedications

This work is dedicated to the memory of our friend and colleague Yuval Hadas, coauthor of [13, 14, 15, 16, 17, 34, 35]. Giorgio Gnecco dedicates his contribution to the memory of his mother Rosanna Merlini.

References

[1] P. Bagnerini, M. Gaggero, G. Gnecco and M. Sanguineti, Diffusion Control of Wildland Fire via a Cooperative Game-Theoretic Model, in *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering (Proceedings of the* 13^{th} *EAI Int. Conf. on Game Theory for Networks*, Springer, 2025 (forthcoming).

[2] F. Biancalani, R. Metulini and G. Gnecco, The Relationship between Players' Average Marginal Contributions and Salaries: An Application to NBA Basketball using the Generalized Shapley Value, *Italian Journal of Applied Statistics*, **35** (2023), 1–29.

[3] D. Braess, Über ein Paradoxon aus der Verkehrsplanung, *Unternehmensforschung*, **12** (1968), 258–268.

[4] D. Braess, A. Nagurney and T. Wakolbinger, On a Paradox of Traffic Planning, *Transportation Science*, **39** (2005), 446–450.

[5] Bureau of Public Roads, Traffic Assignment Manual. U.S. Department of Commerce, Urban Planning Division, 1964.

[6] J. Castro, D. Gómez and J. Tejada, Polynomial Calculation of the Shapley Value based on Sampling, *Computers & Operations Research*, **36** (2009), 1726–1730.

[7] S. Chakravarty, M. Manipushpak and P. Sarkar, *A Course on Cooperative Game Theory*, Cambridge University Press, 2015.

[8] S. Cohen, G. Dror and E. Ruppin, Feature Selection via Coalitional Game Theory, *Neural Computation*, **19** (2007), 1939–1961.

[9] J. R. Correa and N. E. Stier-Moses, Wardrop equilibria, in *Wiley Encyclopedia of Operations Research and Management Science*, John Wiley & Sons, Ltd, 2011, URL https://onlinelibrary.wiley.com/doi/abs/10.1002/9780470400531.eorms0962.

[10] S. Dafermos, Traffic Equilibrium and Variational Inequalities, *Transportation Science*, **14** (1980), 42–54.

[11] B. Gao and Y. Xu, Univariant Approximation by Superpositions of a Sigmoidal Function, *Journal of Mathematical Analysis and Applications*, **178** (1993), 221–226.

[12] M. Girvan and M. Newman, Community Structure in Social and Biological Networks, *Proceedings of the National Academy of Sciences of the USA*, **99** (2002), 7821–7826.

[13] G. Gnecco, Y. Hadas, M. Passacantando and M. Sanguineti, On the Approximation of the Shapley Value via Machine Learning in Transportation Network Cooperative Games, in *AIRO Springer Series - Optimization and Decision Science 2024*, Springer, 2025 (forthcoming).

[14] G. Gnecco, Y. Hadas and M. Sanguineti, Some Properties of Transportation Network Cooperative Games, *Networks*, **74** (2019), 161–173.

[15] G. Gnecco, Y. Hadas and M. Sanguineti, Public Transport Transfers Assessment via Transferable Utility Games and Shapley Value Approximation, *Transportmetrica A: Transport Science*, **17** (2021), 540–565.

[16] G. Gnecco, Y. Hadas and M. Sanguineti, A Game-Theoretic Approach for Reliability Evaluation of Public Transportation Transfers with Stochastic Travel and Waiting Time, *EURO Journal on Transportation and Logistics*, **11** (2022), Article no. 100090.

[17] Y. Hadas, G. Gnecco and M. Sanguineti, An Approach to Transportation Network Analysis via Transferable Utility Games, *Transportation Research Part B - Methodological*, **105** (2017), 120–143.

[18] M. Kendall and J. Gibbons, *Rank Correlation Methods*, 5th edition, Charles Griffin, 1990.

[19] K. Kolykhalova, G. Gnecco, M. Sanguineti, G. Volpe and A. Camurri, Automated Analysis of the Origin of Movement: An Approach Based on Cooperative Games on Graphs, *IEEE Transactions on Human-Machine Systems*, **50** (2020), 550–560.

[20] A. Krylatov, V. Zakharov and T. Tuovinen, *Optimization Models and Methods for Equilibrium Traffic Assignment*, Springer, 2020.

[21] S. Lu, Sensitivity of Static Traffic User Equilibria with Perturbations in Arc Cost Function and Travel Demand, *Transportation Science*, **42** (2008), 105–123.

[22] S. M. Lundberg and S.-I. Lee, A Unified Approach to Interpreting Model Predictions, in *Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS '17)*, Curran Associates Inc., Red Hook, NY, USA, 2017, 4768–4777.

[23] R. Metulini and G. Gnecco, Measuring Players' Importance in Basketball using the Generalized Shapley Value, *Annals of Operations Research*, **325** (2023), 441–465.

[24] T. Michalak, K. Aadithya, P. Szczepański, B. Ravindran and N. Jennings, Efficient Computation of the Shapley Value for Game-Theoretic Network Centrality, *Journal of Artificial Intelligence Research*, **46** (2013), 607–650.

[25] V. Morandi, Bridging the User Equilibrium and the System Optimum in Static Traffic Assignment: A Review, *4OR*, **22** (2024), 89–119.

[26] D. Moretti, F. Patrone and S. Benassi, The Class of Microarray Games and the Relevance Index for Genes, *TOP*, **12** (2007), 256–280.

[27] A. Nagurney and L. Nagurney, The Braess Paradox, in *International Encyclopedia of Transportation* (ed. R. Vickerman), Elsevier, 2021, 601–607.

[28] J. Nash, Non-Cooperative Games, *Annals of Mathematics*, **54** (1951), 286–295.

[29] M. Newman, *Networks: An Introduction*, Oxford University Press, 2010.

[30] A. Nowak and T. Radzik, The Shapley Value for N-Person Games in Generalized Characteristic Function Form, *Games and Economic Behavior*, **6** (1994), 150–161.

[31] M. Osborne and A. Rubinstein, *A Course in Game Theory*, MIT Press, 1994.

[32] B. Panicucci, M. Pappalardo and M. Passacantando, A Path-Based Double Projection Method for Solving the Asymmetric Traffic Network Equilibrium Problem, *Optimization Letters*, **1** (2007), 171–185.

[33] M. Passacantando, Test Networks for the Transportation Network Equilibrium Problem, http://pages.di.unipi.it/passacantando/test_networks.html, Accessed: 2024-12-27.

[34] M. Passacantando, G. Gnecco, Y. Hadas and M. Sanguineti, Braess' Paradox: A Cooperative Game-theoretic Point of View, *Networks*, **78** (2021), 264–283.

[35] M. Passacantando, G. Gnecco, Y. Hadas and M. Sanguineti, On Braess' Paradox and Average Quality of Service in Transportation Network Cooperative Games: A Cooperative Game-Theoretic Point of View, in *Optimization and Decision Science 2020* (eds. R. Cerulli, M. Dell'Amico, F. Guerriero, D. Pacciarelli and A. Sforza), Springer, 2021, 27–37.

[36] M. Passacantando and F. Raciti, Lipschitz Continuity Results for a Class of Parametric Variational Inequalities and Applications to Network Games, *Algorithms*, **16**, Article no. 458.

[37] M. Patriksson, *The Traffic Assignment Problem: Models and Methods*, Courier Dover Publications, 2015.

[38] B. Peleg and P. Sudhölter, *Introduction to the Theory of Cooperative Games*, Kluwer Academic Publishers Group, 2007.

[39] F. Qayyum, N. Abdel Samee, M. Alabdulhafith, A. Aziz and M. Hijjawi, Shapley-based Interpretation of Deep Learning Models for Wildfire Spread Rate Prediction, *Fire Ecology*, **20** (2024), 1–21.

[40] T. Roughgarden, On the Severity of Braess's Paradox: Designing Networks for Selfish Users is Hard, *Journal of Computer and System Sciences*, **72** (2006), 922–953.

[41] T. Roughgarden and E. Tardos, Introduction to the Inefficiency of Equilibria, in *Algorithmic Game Theory* (eds. N. Nisan, T. Roughgarden, E. Tardos and V. Vazirani), Cambridge University Press, 2007, 443–469.

[42] L. Shapley, A Value for N-Person Games, in *Contributions to the Theory of Games* (eds. H. Kuhn and A. Tucker), vol. 28 of Annals of Mathematical Studies, Princeton University Press, 1953, 307–317.

[43] M. Smith, The Existence, Uniqueness and Stability of Traffic Equilibria, *Transportation Research Part B: Methodological*, **13** (1979), 295–304.

[44] A. Smola and B. Schölkopf, A Tutorial on Support Vector Regression, *Statistics and Computing*, **14** (2004), 199–222.

[45] J. Wardrop, Some Theoretical Aspects of Road Traffic Research, *Proceedings of the Institution of Civil Engineers*, **1** (1952), 325–362.