IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received May 24, 2020, accepted June 1, 2020, date of publication June 11, 2020, date of current version June 24, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3001676

Revealing the Character of Nodes in a Blockchain With Supervised Learning

RADOSŁAW MICHALSKI[1], DARIA DZIUBAŁTOWSKA[2], AND PIOTR MACEK[1]
[1]Department of Computational Intelligence, Wrocław University of Science and Technology, 50-370 Wrocław, Poland
[2]Faculty of Mathematics, Wrocław University of Science and Technology, 50-370 Wrocław, Poland

Corresponding author: Radosław Michalski (radoslaw.michalski@pwr.edu.pl)

This work was supported by the statutory funds of the Faculty of Computer Science and Management, Wrocław University of Science and Technology.

ABSTRACT The term blockchain has its roots in cryptocurrencies. However, its applications are now more widespread, and in many areas, this technology has become the foundation of the distributed ledger. The blockchain protocol assumes that all the participants of the system are both contributors and safeguards of this ledger, since the lack of a trusted third party requires other security precautions in order to maintain the consistency of transactions. In this work, we investigate whether for the participants of a blockchain-based system that does not require revealing the character explicitly, it can be discovered by other means. In order to verify this, we built and publicly released a dataset of nearly 9,000 addresses of nodes in the most popular cryptocurrency - Bitcoin, and then labelled them. These labels represent the character the nodes have in the network, e.g. miners or exchanges. We then developed a set of features that quantify the behaviour of nodes in the network and used supervised machine learning algorithms to find out whether the character of nodes can be revealed based on these features. Our results demonstrate, due to the F-score reaching over 95% in the best-performing algorithms, that it is hard to hide the role the node has in a blockchain-based network. These results indicate that to build trustworthy blockchain-based systems that fully comply with original blockchain assumptions, specific countermeasures are needed in order to preserve the desired level of anonymity.

INDEX TERMS Blockchain, supervised learning, machine learning, Bitcoin, cryptocurrencies, node2vec.

I. INTRODUCTION

Nowadays, blockchain, as a trusted transaction register, supports multiple application areas. In many of these applications, the entities are easily identifiable and, since they perform a series of connected activities in the business process, have specific roles assigned. For instance, this is the case for supply chain solutions. On the other hand, in certain areas, such as most popular cryptocurrencies, despite full transparency of the transaction register itself, many entities try to keep their pseudonymity regarding both identity and the role they play in the process. As the blockchain of Bitcoin and many other cryptocurrencies follows this protocol, its users are not required to reveal any details. It is only in the case of linkage to fiat currencies that some exchanges require the *know your customer* (KYC) procedure to be carried out. Otherwise, a legitimate Bitcoin blockchain node can stay at the level of only exposing publicly its alphanumeric address. Even if forced to reveal the details due to KYC, these are only shared with a single entity.

However, this does not necessarily mean that it is not possible at least to reveal the role of a particular node in the blockchain ecosystem of cryptocurrencies if this node performs some activities. It is linked to the fact that these activities are usually associated with the role of a node. Independently, whether we are considering exchanges, mining pools, miners, businesses or individuals, when analysing the blockchain as a complex network linking entities with transactions, we demonstrate that it can be possible to reveal to which role a particular node belongs.

In this work, we propose and evaluate supervised machine learning approaches for classifying nodes to the roles in a Bitcoin blockchain. In order to demonstrate the capabilities of this approach, we built a dataset of nearly 9,000 Bitcoin addresses and labelled them with their most plausible role. Afterwards, we developed a set of features used for

The associate editor coordinating the review of this manuscript and approving it for publication was Hong-Ning Dai.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

classification and evaluated these using different supervised learning algorithms, including decision trees, KNN, kernel-based SVM or random forest. We also evaluated the performance of these methods when using node2vec encoding of node features.

The results indicate that nodes that perform transactions reveal details that can be used for classifying them into specific roles. It means that the nodes that would like to achieve a certain level of anonymity in a blockchain-based system should introduce countermeasures that decrease the risk of revealing their character in the ecosystem. The results of this study can lead either to individual actions of blockchain nodes or the incorporating of risk mitigation mechanisms into systems that are based on the blockchain increasing their trustworthiness.

The next section presents the related research in this area, and Section III presents the experimental setting. In Section IV, we share the results of this study and Section V contains a discussion on these results. In the last section, we summarise the work and indicate potential future research directions. We also made the dataset used in this study publicly available.

II. RELATED RESEARCH

As it is hard to claim that solutions based on blockchain have entered a mature stage, the research in this area is taking many different directions, as described in [31]. On the one hand, a significant amount of research is devoted to the security of blockchain-based systems. This is mainly due to the fact that in many cases, the well-known 51% majority attack still poses a risk to the integrity of the whole ecosystem [10], [11], [23]. Another important research direction in terms of security is the fork problem, which is leading to the emergence of a rivalling sibling of a given blockchain [25]. These two risks are mainly for cases of systems with open participation. However, for some industrial applications, the risk is still not negligible.

The problem tackled in this work can be considered as another related to the security of blockchain-based systems where transactions are seen as transparent to all the nodes. Here, we are interested in revealing the character a node has based on a set of features that can be derived from the ledger itself. Obviously, in some applications, the character of a node can be made public and become a part of the framework, but for others, it might not be required to reveal it. Hence, in this work, we investigate how well certain features and machine learning algorithms perform in this task. In general, this research direction, when applied to other domains, is not a novel one, and machine learning is often used in combination with other networks. More formally, one can consider this problem as a multiclass classification problem, which has been applied to networks in a number of areas presented below.

For instance, based on the ideas presented in [8], the researchers used features of a social network of co-arrestees for predicting the possibility of future violent crimes [27]. A similar concept was also used in [29] for analyzing co-offending networks. In this work, a co-offence prediction algorithm using supervised learning has been developed - by using features derived from the network, it was possible to nominate co-offenders. Nevertheless, the classification in networks based on communication or behaviour in social media can relate to completely different areas, such as poverty detection [28], personality traits discovery [22] or occupation [17]. Similarly, machine learning is used to reveal the position in an organizational structure based on activity in a corporate e-mail exchange [26]. All this is possible because our digital traces differ depending on our role or status.

It should also be noted that the approaches used for classifying nodes can either treat the instances independently based on the assumption of independence and identical distribution of instances, as studied in [29] or consider the problem as a relational classification by proposing collective classification methods [20], [26]. In this work, we follow the former approach, but collective classification can be considered as an interesting future work direction.

This study can be perceived as belonging to the family of studies that investigate the possibilities of the deanonymization of nodes in cryptocurrency networks. This family tackles the problem at different levels, e.g., by identifying the IP addresses of nodes [19], or by studying transaction motifs analysis [18] or transaction history summarization [24]. This direction was summarized in [1]. Some other works also tackled the problem of identifying the nodes based on selected features. For instance, authors of [6] looked at how the importance of nodes calculated by the PageRank measure correlated with the type of a node in the Ethereum blockchain. Apart from this aspect, authors investigated the distributions of other measures in the network built upon transactions showing that they follow the power law. This direction what was also followed up in [12]. The capabilities of machine learning in the area of blockchain have been evaluated to discover Ponzi schemes either for Ethereum [7] or Bitcoin [4] - this could be considered as a one-class classification problem. Our work extends the area by demonstrating how multiple supervised machine learning algorithms perform in the area of deanonymizing node types when using a large dataset that has been collected for evaluation and made public. Moreover, we look at whether and how node2vec embedding contributes to classification accuracy. The outcome of research works in this area can be used for improving the design of blockchain-based trustworthy systems.

III. EXPERIMENTAL SETTING

This section contains information on the experimental setting, including the dataset description, features, classification methods and evaluation metrics.

A. DATASET

The Bitcoin ecosystem provides a generic way of exchanging cryptocurrency. This imposes that the entities making transactions can be of any type. However, due to the nature of

the blockchain protocol and the way it interacts with external entities, some typical roles can be distinguished. These roles, however, are not announced as a part of the protocol, so they can only be discovered by other means. In this research, we focused on the following roles: mining pools, miners, coinjoin services, gambling, exchanges and services. These roles are described below.

- Mining pools — these gather miners in order to maximize their chances for mining new block by combining computational power. If the block will be mined by this particular mining pool, users contributing to it gain their qdequate share to the computational resources provided.
- Miners — each node in the protocol has the option either to not mine new blocks or to do so. In the latter case, it can join the mining pool described above or do it independently. In this research, we focused on the miners joining mining pools, and have been identifying them based on the transactions that follow the mining of new blocks by mining pools. The criteria to identify particular addresses as miners were the following: in order to the source address of a transaction to be identified as a mining pool, the destination addresses had to be more than ten and each address had to gain less than 1 BTC.
- Coinjoin–like, gambling, exchanges, services — addresses assigned to these categories were obtained with the help of the service walletexplorer.com. This website collects sets with addresses for different services, websites, wallets, exchanges, etc. The content of this website is searched by each of the addresses and if the name of any service is found in the wallet explorer database, a proper category is assigned based on the owner found. Labels for some addresses found on this site, categorized as ''services/other'', were divided between services and coinjoin-like. In the category *services*, the most frequently found labels were wallets.

The list of the classes contains the most typical ones to be found in Bitcoin blockchain, but if one will be able to identify new types of entities, the list could be extended or modified.

Our initial set of addresses to evaluate and assign labels based on the Bitcoin blockchain blocks from 520,890 to 520,910. These blocks have been downloaded and parsed by our tools, but researchers interested in the datasets that would like to use already published datasets for different cryptocurrencies are able to find such data for selected ones, such as Ethereum [32] or Bitcoin [21]. This data can then undergo labelling that we performed in our work, as described below.

We performed the identification of about 8,800 addresses, and information on the on the number of addresses belonging to each class we considered is presented in Table 1. In order to extend the number of cases from underrepresented classes, our analysis had to consider also other blocks than the ones listed above, since we did not find many instances of these. In this case, we looked for the addresses that have been previously identified as belonging to these classes or have been publicly announced by their owners as a part of their operations. The labelling of addresses has been done by crawling and scrapping information published on the websites of services, posts on variety of forums and other sources. It is worth underlining that there is no guarantee that the tagging of these addresses is error-prone, but this is unavoidable, since the Bitcoin protocol does not consider linking the addresses to types of entities. The dataset containing the labels of addresses has been made public and is available at Harvard Dataverse.[1]

TABLE 1. Numbers of instances for identified addresses.

Label	Number of instances
mining pools	89
miners	4,030
coinjoins	800
gambling	911
exchanges	1,666
services	1,312
total	8,808

B. FEATURES

By having a list of Bitcoin addresses with corresponding labels, we retrieved a list of all transactions for all addresses from the blockchain. We chose two different approaches, based on transactions, to calculate the features for an address:

1) based on the address's statistics, and solely based on transactions that the investigated address was part of. This approach offers a lot flexibility, as all the address's transactions are used to calculate features. The full list of 149 features can be found in Table 7. Features can be grouped into the following families:
   - numbers of addresses in transaction inputs and outputs
   - amounts of bitcoins sent in transactions
   - numbers of coinbase transactions
   - numbers of transactions
   - lifetime (in blocks)
   - address type (Pay to Script Hash, Pay To Public Key Hash etc.)
2) address embeddings calculated using the node2vec [14] algorithm. In order to be able to calculate address embeddings, we treat addresses as nodes in a graph, and transactions between them as edges, as presented in Figure 1. This approach requires a transaction graph to be created that is based on all transactions in a selected range of blocks, and not only for the investigated addresses. Therefore, it is computationally hard to include all transactions which the investigated address was part of. Most addresses have transactions in a large range of blocks, depending on how long the address was in use, which implies the need to compute a large transaction network. The algorithm calculates embeddings for all addresses in a selected range of blocks, and we only select embeddings for addresses that are part of our dataset. Because of memory constraints, we

[1] https://doi.org/10.7910/DVN/KEWU0N

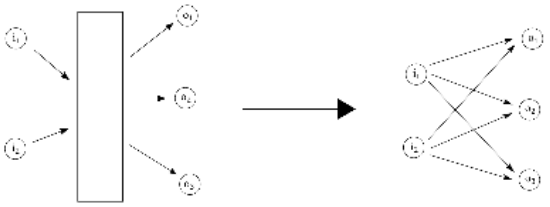

FIGURE 1. An example of creating a graph from a single transaction.

selected 100 blocks with numbers ⟨520850; 520950⟩ from the Bitcoin blockchain and calculated address embeddings based on the transaction graph found in those blocks. The selected blocks contain the majority, although not all, of the addresses in the dataset.

We selected the following parameter values for the node2vec algorithm:

- embedding vector dimension: 128
- number of iterations: 1,000
- walk length: 80
- number of walks per source: 10

C. EVALUATED SUPERVISED LEARNING ALGORITHMS

Machine learning is an art of programming where computers can learn from given data without being precisely programmed. The two most essential types of machine learning methods are supervised learning and unsupervised learning. In this work, we focused on the first one, in which an algorithm is learning from data and is provided with the correct solution (labels). The following supervised learning methods were evaluated: Decision Trees, Random Forests, Extra Trees, Neural Network, SVM, Logistic Regression, and k-nearest Neighbours. Apart from the typical approach based on features, we looked at how the node2vec embedding technique contributes to the results when used with the aforementioned classification approaches.

The Decision Tree algorithm is a process of sequential decisions, which starts from a *root*, and after making a decision based on the given rule it spreads out, creating a new *branch* and lead down to the left or right node [30]. At the bottom of this structure there are *leafs*, which contain predicted classes for decision paths that lead from the root to the corresponding leaf. The Decision Tree algorithm looks for a rule that can decrease the impurity measure the most.

Random Forests algorithms use a technique called Ensemble Learning [5]. It is based on the assumption that aggregating predictions of a group of classifiers often provides better predictions. In Random Forests, the classification algorithms used for this purpose are Decision Trees. Many of them are trained on randomly chosen subsets of all data features. New instances are classified by the majority of votes. Random Forests, opposed to Decision Trees, do not provide clear information on why some predictions were made in precisely this way, and what specifically contributed to this choice. One advantage of the Random Forests algorithm, which will be used later in this paper, is the possibility to determine the importance of each feature in a given dataset.

The Extra Trees method is a variation of Random Forests, with the difference being that each tree is trained on the whole set of training data instead of on its subsets [13]. This leads to the minimizing of bias. Additionally, the cut-point is chosen randomly, which along with the first difference, should be able to reduce variance. Moreover, the Extreme Trees algorithm has better computational performance than the standard RF model.

The neural network is an attempt to imitate how the brain works [3]. The Neural Network model consists of layers that are built with artificial neurons connected to each other. A neuron can have multiple input values passed from the previous layer (or input values in the case of the input layer). Received signals are multiplied by weights, which can be adjusted during the training process, and used as an input of the neuron's activation function. An output layer usually has a number of neurons that is equal to the number of classes. For net training, back-propagation is used, which tries to minimize the cost function. Due to its non-linearity, learning is done in the iterative process, which can result in high time complexity of an algorithm. The Neural Networks model, due to its architecture, is more difficult to interpret, and it cannot provide any more insights on the dataset.

Logistic Regression is a classifier that works in a similar way to linear regression. It was first designed for binary cases. However, it can be easily extended to multiclass problems. For each class, a separate binary classifier is created, and therefore for N-classes, there will be N binary classifiers - one for each class [16]. Logistic Regression is a probabilistic model that evaluates the probability of being a member of a given class. It uses the maximum likelihood method. Logistic Regression can overfit in high dimensional datasets, and to avoid this L1 and L2 techniques should be used. The algorithm is simple to implement and easy to understand.

SVM, similar to Logistic Regression, solves the binary classification problem, which can be extended for multiclass classification [9]. However, it is not a probabilistic model. For N number of features, it divides N-dimensional space with hyperplanes that look for the maximum distance between datapoints from the training set. New objects are classified based on which side of the hyperplanes they are placed. As opposed to Linear Regression, this method is effective for high dimensional data.

K-nearest neighbours is a non-parametric model without any built-in statistical model or function describing data distribution [2]. In KNN a new object becomes a part of the most common class among its k neighbours. If $k = 1$, then objects are assigned to the class of its only one neighbour. KNN is one of the simplest classification methods that does not require a training process.

Classification algorithms were tuned by selecting the best hyper-parameter values, which are listed in Table 2.

D. EVALUATION

To evaluate the classification results and to measure quality we used the Macro-F1 score and confusion matrix. We also

TABLE 2. Parameters tested for each type of model for the proposed set of features based on the transaction list. The chosen values that yielded the best results are in bold.

Parameter name	Searched values
SVM	
C parameter	1, 10, 100, **1000**
kernel	**linear**, RBF
Neural Network	
hidden layers size	(100,), **(150, 30)**, (30,), (10,), (20, 20)
activation function	**ReLU**
optimization method	**Adam**
Decision Trees	
class weight	balanced, **balanced subsample**, uniform
criterion	**entropy**, Gini
Random Forest	
class weight	balanced, **balanced subsample**, uniform
criterion	**entropy**, Gini
maximum depth	10, 15, **20**, 30, 50
maximum number of features	auto, log2, **35**
number of classifiers	10, 100, **200**, 1000
Extra Trees	
class weight	**balanced**, balanced subsample, uniform
criterion	**entropy**, Gini
maximum depth	10, **15**, 20, 30, 50
number of classifiers	10, **100**, 200, 1000
Logistic Regression	
regularization parameter	1, 10, 100, **1000**
class weight	**balanced**, uniform
penalty	**L1**, L2
KNN	
K	**3**, 5, 11, 15, 21
elements weights	**distance**, equal

evaluated the classification results using subsets of the most important features. We used a built-in feature of the Random Forest algorithm to calculate feature importance.

We also evaluated how the number of transactions used to calculate an address's features affects the classification results. This experiment allows us to estimate how many transactions an address has to be part of in order for it to be able to be classified with enough confidence.

All the experiments were performed using repeated stratified K-fold with 10 folds, and repeated five times.

The classification algorithms were tuned by evaluating different configurations of hyper-parameter values in order to find the configuration which gives the best results.

Categorical variables are encoded using one-hot vectors, and different preprocessing techniques were used for numeric variable preprocessing [15]:

- no preprocessing, use values as-is
- unit-length normalization, performed separately for each sample. Each vector is transformed so that it is a unit vector of length 1.
- feature standardization (Z-score normalization), performed separately for each feature. Standardization makes the values have zero-mean and unit variance.
- min-max normalization, re-scales feature values to range $\langle 0; 1 \rangle$.

IV. RESULTS

This section presents the experimental results from a number of perspectives. Firstly, we look at the overall performance of the evaluated algorithms and the influence of preprocessing methods on the results - this is described in Section IV-A. Next, in Section IV-B, we evaluate how the number of features used for classification impacts the quality of classification. Another perspective on the subject is presented in Section IV-C, which was the answering of the question of how much historical data about the nodes' transaction is required to assign labels accurately. Lastly, we investigate how a relatively new technique of using node2vec vector embeddings performs when being an input for classifiers (Section IV-D).

A. THE OVERALL PERFORMANCE OF SUPERVISED LEARNING TECHNIQUES

In this section, we present the overall performance of supervised learning techniques. These results are summarized in Table 3. The results are also decomposed by the initial data preprocessing techniques - the best-performing preprocessing technique for a given supervised method is presented in bold.

One can clearly see that the overall performance for the evaluated methods varies, but, on average the Macro-F1 score is about 0.8, with notable exceptions of Random Forest - 0.96, and Decision Trees - 0.91. Logistic regression, even if not the best performing approach, was least sensitive to the preprocessing methods used.

B. FEATURES SELECTION

The performance of Decision Trees is not affected by a high number of features, as they have the ability to measure their importance. However, it can cause problems with the generalization of a model that is created during the training process, which can in turn lead to worse results. The performance of algorithms which are not built upon Decision Trees: *MLP*, *SVM*, *Logistic Regression*, and *KNN* were tested with a lower number of features included. Features were chosen based on their importance, which was evaluated with the Decision Tree CART algorithm. Table 4 and Figure 2 present the classification results of the selected methods with a varuable number of the most important features. These show that the number of features used in the experiment did not have any impact on the achieved scores in any way. The decline can only be observed in the case of a small number of features $n = 10$ bein taken under consideration. For greater n, the differences are small. Classification with all the features used gave the best results for the neural network (*MLP*) and SVM. Nevertheless, after removing some less important features, logistic regression and KNN achieved the best F-scores. The described results were conducted with using the models' hyper-parameters described in the previous section.

Ten out of 149 features with the highest importance for the Random Forest algorithm are presented in Figure 3.

C. INFLUENCE OF THE NUMBER OF TRANSACTIONS ON CLASSIFICATION RESULTS

The number of transactions which the given address was a part of has a significant impact on describing its features.

TABLE 3. The classification performance of the evaluated supervised learning techniques in the nodes' types classification task. The evaluation measure was the F-score and the best results for particular methods for different types of data preprocessing are in bold.

	Decision Trees	Extra Trees	KNN	SVM	Logistic Regression	Neural Network	Random Forest
MinMax	0.794	0.760	0.733	0.642	0.713	0.716	0.859
Normalization	0.704	0.706	0.717	0.727	0.713	0.712	0.771
Scaling	**0.910**	0.759	**0.855**	**0.790**	**0.719**	**0.850**	0.944
No scaling	0.906	**0.860**	0.683	0.423	0.718	0.386	**0.960**

TABLE 4. F-score results for the tested classifiers with a different number of features included based on their importance.

Number of features	KNN	SVM	Logistic Regression	MLP
10	0.786	0.389	0.560	0.701
15	0.829	0.509	0.617	0.796
20	0.856	0.577	0.674	0.811
30	**0.863**	0.598	0.690	0.812
50	0.860	0.596	0.687	0.811
70	0.859	0.600	0.683	0.812
90	0.859	0.581	0.685	0.810
110	0.861	0.611	**0.691**	0.809
149	0.800	**0.760**	0.650	**0.850**

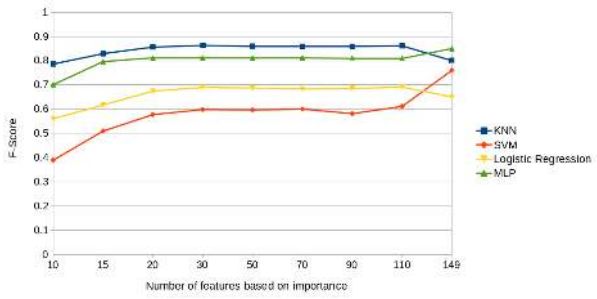

FIGURE 2. F-score results for the tested classifiers with a different number of features included based on its importance.

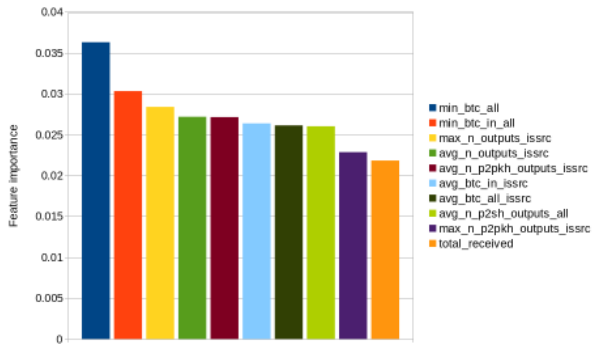

FIGURE 3. Top ten of the most important features.

That is why the addresses with a small or relatively high number of transactions can cause issues for classification algorithms. In this experiment, the number of N first transactions of a given address was used for the evaluation of feature values. The following N values were chosen: $N \in \{1, 3, 5, 7, 10, 20, 50, 100, 200, 500, 1000, 5000, 10000\}$.

We split the addresses into ten folds according to stratified cross-validation rules. Nine of them were used as training examples and their attributes were evaluated based on the whole transaction history. The next step was to evaluate the

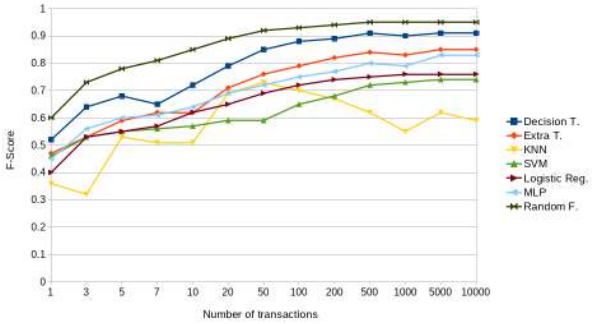

FIGURE 4. F-score results for the chosen classifiers evaluated with N-first transactions for each address in the test dataset.

features for test addresses for each N of transactions taken under consideration.

With this approach, we were able to study the dependency between classification results and the number of transactions used to calculate the measures of addresses from the test set. Results for classification for different models and N values are presented in Figure 4.

There is no visible connection between the number of transactions used and the score achieved by the KNN algorithm, which is most probably due to its unpredictability and dependence on how the data was divided. Other algorithms provide better results with an increase of N. Based on Figure 4, we can assume that only 500 transactions need to be used for feature evaluation in order to achieve the highest possible score. Even the classification quality with only 100-200 transactions is not much lower than the best scores. Figure 5 shows a confusion matrix for the Random Forest algorithm with $N = 100$. It can be observed that most of the classes were correctly assigned. Again, the classification of *services* poses a challenging task, since objects of this class are incorrectly labelled as *miners*.

D. RESULTS FOR node2vec VECTOR EMBEDDINGS

We evaluated the supervised learning algorithms listed in Section III-C with vector embeddings used as address features. Embedding vectors are calculated using the node2vec algorithm and have a fixed length, so it was not possible to examine feature importance or the impact of a number of address transactions. Classification algorithms were tuned by selecting the best hyper-parameter values, which are listed in Table 5.

The best results were achieved using algorithms that did not yield the best results when using custom features based on transaction history. Moreover, the decision-tree based algo-

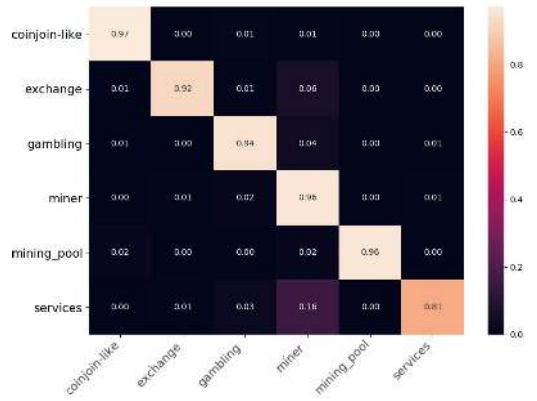

FIGURE 5. Confusion matrix for the random forest algorithm on a test dataset created with the first 100 transactions for each address in the test dataset.

TABLE 5. Parameters tested for each type of model for node2vec embedding vectors. Chosen values which yielded best results are in bold.

Parameter name	Searched values
SVM	
C parameter	1, 10, 100, **1000**
kernel	**linear**, RBF
Neural Network	
hidden layers size	(100,), **(150, 30)**, (30,), (10,), (20, 20)
activation function	**ReLU**
optimization method	**Adam**
Decision Trees	
maximum depth	**10**, 15, 20, 30, 50
criterion	**entropy**, Gini
Random Forest	
class weight	balanced, **balanced subsample**, uniform
criterion	**entropy**, Gini
maximum depth	**10**, 15, 20, 30, 50
number of classifiers	10, 100, **200**, 1000
Extra Trees	
criterion	entropy, **Gini**
maximum depth	**10**, 15, 20, 30, 50
number of classifiers	10, **100**, 200, 1000
Logistic Regression	
regularization parameter	**1**, 10, 100, 1000
penalty	**L1**, L2
KNN	
K	**3**, 5, 11, 15, 21
elements weights	**distance**, equal

rithms yielded lower scores than for custom features, which may be a consequence of how decision trees work. The SVM algorithm gave the best F-score value of 91%, followed by Multi-Layer Perceptron, which yielded an F-score result of 90%. All the results are presented in Table 6.

Figure 6 presents ą normalized confusion matrix for the results of the SVM algorithm. In this figure, it is clearly visible that addresses from majority classes are best predicted, and other classes with fewer samples are often wrongly assigned to one of the majority classes, usually the 'miner' class. This is most likely a result of an unbalanced dataset - the majority class samples dominate most of the vector space, which makes it difficult for classification algorithms to generalize minority classes sufficiently.

TABLE 6. Table presenting F-score results for address classification based on node2Vec embedding vectors.

Dec. T.	Ex. T.	KNN	SVM	Log. Reg.	MLP	Rnd. F.
0.62	0.61	0.81	**0.91**	0.79	0.90	0.78

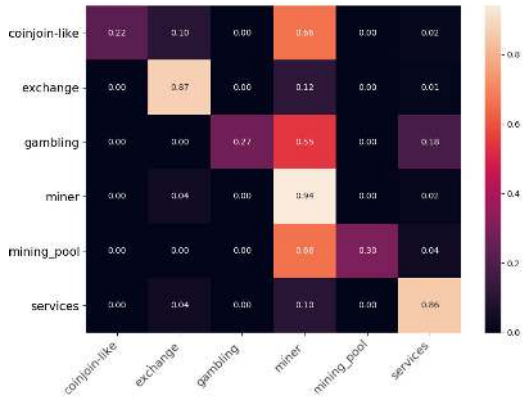

FIGURE 6. Normalized confusion matrix for results of the SVM algorithm using node2Vec embeddings.

V. DISCUSSION

The results indicate that the set of features built and later used by supervised learning algorithms demonstrates that it is possible to discover the character of a node in a blockchain-based cryptocurrency system. The algorithms based on Decision Trees generally result in a high F-score. The best performing ensemble algorithm was Random Forest, reaching an F-score of 95%. This means that the proposed features are able to map the behaviour of nodes belonging to different classes. Another experiment showed that the embeddings of nodes used as an input for classifiers also provide good results, but at the same time it is worth noting that all the algorithms based on decision trees performed worse with this approach. The best performance in terms of F-score in this setting was achieved by SVM - 91%. However, when looking into the details, it was observed that the classifier improperly assigned the majority class to the instances of underrepresented classes. This behaviour is not observed in the case of classification with the set features proposed by us.

Another set of experiments also showed that depending on the algorithm, the F-score varies when the least features are being provided. Whilst SVM performed better when receiving more features, KNN and logistic regression reduced their performance when they were based on too few or too many features.

As the nodes perform the activity in the network, they tend to reveal more details about their behaviour. However, the F-score dependence on the number of transactions the nodes participated in is surprisingly sublinear, and therefore even for a limited number of transactions, one can derive a fair amount of information about which classes the nodes belong to. For instance, the best-performing algorithm, Random Forest, only requires seven transactions to reach an F-score of 80%.

TABLE 7. Table presenting a summarized list of all the proposed features based on an address's transaction history.

Description of features
• mean/min/max/total number of transaction inputs, when the address in question is an input/output of the transaction • mean/min/max/total number of transaction outputs, when the address in question is an input/output of the transaction • mean/min/max/total number of Pay To Script Hash addresses on transaction input, when the address in question is an input/output of the transaction • mean/min/max/total number of Pay To Public Key Hash on transaction input, when the address in question is an input/output of the transaction • mean/min/max/total number of Pay To Script Hash addresses on transaction output, when the address in question is an input/output of the transaction • mean/min/max/total number of number of Pay To Public Key Hash on transaction output, when the address in question is an input/output of the transaction • mean/min/max/total number of unique Pay To Script Hash addresses, with whom the address in question was part of the same transaction • mean/min/max/total number of unique Pay To Public Key Hash, with whom the address in question was part of the same transaction • mean/min/max/total amount of BTC sent by Pay To Script Hash/Pay To Public Key Hash addresses, when the address in question is an input/output of the transaction • mean/min/max/total amount of BTC in transactions, when the address in question is an input/output of the transaction • mean/min/max/total amount of BTC on transaction input, when the address in question is an input/output of the transaction • mean/min/max/total amount of BTC on transaction output, when the address in question is an input/output of the transaction • mean/min/max/total amount of BTC in transactions • mean/min/max/total amount of BTC on transaction input • mean/min/max/total amount of BTC on transaction output • number of coinbase transactions • number of unique input addresses, in transactions when the address in question is an input/output of transaction • number of unique output addresses, in transactions when the address in question is an input/output of transaction • number of unique addresses, with whom the address in question was part of the same transaction • number of transactions for the address • number of transactions, where the address in question was input • number of transactions, where the address in question was output • number of transactions, where one of the input addresses is also one of output addresses (change output) • address type (Pay to Script Hash, Pay To Public Key Hash etc.) • lifetime (range of blocks between which the address in question had all its transactions)

VI. CONCLUSION

The primary purpose of this work was to investigate whether it is possible to reveal the character of nodes in a blockchain-based network based on a set of features built upon transactions history and supervised learning algorithms. The reason behind this study was to find out whether blockchain-based systems are prone to deanonymization in certain aspects, and the obtained results tend to confirm this claim. Not every blockchain follows the same principles and protocol as the one based on Bitcoin. However, by conducting this study, we would like to underline that these results should be taken into account in the case of designing trustworthy blockchain-based systems, since any wrong design decisions can lead to undesired consequences regarding the privacy of their participants.

One of the most interesting future research directions is related to verifying the proposed approach against a non-cryptocurrency related blockchain. Another direction can relate to applying collective classification techniques, such as Loopy Belief Propagation or Iterative Classification. One can also look for other classes of addresses, e.g. by identifying scamming behaviour. Moreover, the feature set can be extended by the features related to network measures from the graph built upon transactions, as in selected works presented in related research these measures appeared to be helpful in manual identification of types of entities. However, due to the size of many blockchains, it is anticipated that a method limiting the number of blocks considered will be required, as some measures are computationally complex. This could be done either by selected methods of sampling, for instance snowball sampling using transaction addresses or simply by limiting the number of blocks from the blockchain. However, the former method can result with higher connectivity in the graph.

For researchers interested in further exploring this direction and willing to compare their approaches against ours, we made the dataset with labels public.

REFERENCES

[1] N. Alsalami and B. Zhang, "SoK: A systematic study of anonymity in cryptocurrencies," in *Proc. IEEE Conf. Dependable Secure Comput. (DSC)*, Nov. 2019, pp. 1–9.
[2] N. S. Altman, "An introduction to kernel and nearest-neighbor nonparametric regression," *Amer. Statistician*, vol. 46, no. 3, pp. 175–185, Aug. 1992.
[3] T. O. Ayodele, "Types of machine learning algorithms," in *Proc. New Adv. Mach. Learn.*, 2010, pp. 19–48.
[4] M. Bartoletti, B. Pes, and S. Serusi, "Data mining for detecting bitcoin ponzi schemes," in *Proc. Crypto Valley Conf. Blockchain Technol. (CVCBT)*, Jun. 2018, pp. 75–84.
[5] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.
[6] T. Chen, Y. Zhu, Z. Li, J. Chen, X. Li, X. Luo, X. Lin, and X. Zhange, "Understanding ethereum via graph analysis," in *Proc. IEEE Conf. Comput. Commun.*, Apr. 2018, pp. 1484–1492.
[7] W. Chen, Z. Zheng, J. Cui, E. Ngai, P. Zheng, and Y. Zhou, "Detecting ponzi schemes on ethereum: Towards healthier blockchain technology," in *Proc. World Wide Web Conf. World Wide Web - WWW*, 2018, pp. 1409–1418.
[8] N. Coles, "It's not what you know' it's who you know that counts. Analysing serious crime groups as social networks," *Brit. J. Criminol.*, vol. 41, no. 4, pp. 580–594, 2001.
[9] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, 1995.
[10] D. Dasgupta, J. M. Shrein, and K. D. Gupta, "A survey of blockchain from security perspective," *J. Banking Financial Technol.*, vol. 3, no. 1, pp. 1–17, Apr. 2019.
[11] S. Dey, "Securing majority-attack in blockchain using machine learning and algorithmic game theory: A proof of work," in *Proc. 10th Comput. Sci. Electron. Eng. (CEEC)*, Sep. 2018, pp. 7–10.
[12] S. Ferretti and G. D'Angelo, "On the ethereum blockchain structure: A complex networks theory perspective," *Concurrency Comput., Pract. Exper.*, vol. 32, no. 12, Jun. 2020.
[13] P. Geurts, D. Ernst, and L. Wehenkel, "Extremely randomized trees," *Mach. Learn.*, vol. 63, no. 1, pp. 3–42, Apr. 2006.
[14] A. Grover and J. Leskovec, "Node2vec: Scalable feature learning for networks," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2016, pp. 855–864.
[15] J. Han, J. Pei, and M. Kamber, *Data Mining: Concepts Techniques*. Amsterdam, The Netherlands: Elsevier, 2011.
[16] S. Lemeshow and R. X. Sturdivant, *Applied Logistic Regression*, vol. 3980. Hoboken, NJ, USA: Wiley, 2013.

[17] Y. Huang, L. Yu, X. Wang, and B. Cui, "A multi-source integration framework for user occupation inference in social media systems," *World Wide Web*, vol. 18, no. 5, pp. 1247–1267, Sep. 2015.
[18] M. Jourdan, S. Blandin, L. Wynter, and P. Deshpande, "Characterizing entities in the bitcoin blockchain," in *Proc. IEEE Int. Conf. Data Mining Workshops (ICDMW)*, Nov. 2018, pp. 55–62.
[19] P. L. Juhász, J. Stéger, D. Kondor, and G. Vattay, "A Bayesian approach to identify bitcoin users," *PLoS ONE*, vol. 13, no. 12, Oct. 2018, Art. no. e0207000.
[20] T. Kajdanowicz, R. Michalski, K. Musial, and P. Kazienko, "Learning in unlabeled networks—An active learning and inference approach," *AI Commun.*, vol. 29, no. 1, pp. 123–148, Oct. 2015.
[21] D. Kondor, I. Csabai, J. Szäle, M. Pósfai, and G. Vattay, "Inferring the interplay between network structure and market effects in bitcoin," *New J. Phys.*, vol. 16, no. 12, Dec. 2014, Art. no. 125003.
[22] M. Kosinski, D. Stillwell, and T. Graepel, "Private traits and attributes are predictable from digital records of human behavior," *Proc. Nat. Acad. Sci. USA*, vol. 110, no. 15, pp. 5802–5805, Apr. 2013.
[23] I. C. Lin and T. C. Liao, "A survey of blockchain security issues and challenges," *Int. J. Netw. Secur.*, vol. 19, no. 5, pp. 653–659, 2017.
[24] Y.-J. Lin, P.-W. Wu, C.-H. Hsu, I.-P. Tu, and S.-W. Liao, "An evaluation of bitcoin address classification based on transaction history summarization," in *Proc. IEEE Int. Conf. Blockchain Cryptocurrency (ICBC)*, May 2019, pp. 302–310.
[25] T. Neudecker and H. Hartenstein, "Short paper: An empirical analysis of blockchain forks in bitcoin," in *Proc. Int. Conf. Financial Cryptography Data Secur.* Cham, Switzerland: Springer, 2019, pp. 84–92.
[26] M. Nurek and R. Michalski, "Combining machine learning and social network analysis to reveal the organizational structures," *Appl. Sci.*, vol. 10, no. 5, p. 1699, Mar. 2020.
[27] E. Shaabani, A. Aleali, P. Shakarian, and J. Bertetto, "Early identification of violent criminal gang members," in *Proc. 21th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2015, pp. 2079–2088.
[28] P. Sundsáy, J. Bjelland, B.-A. Reme, A. M. Iqbal, and E. Jahani, "Deep learning applied to mobile phone data for individual income classification," in *Proc. Int. Conf. Artif. Intell., Technol. Appl.*, 2016, pp. 1–10.
[29] M. A. Tayebi, M. Ester, U. Glässer, and P. L. Brantingham, "Spatially embedded co-offence prediction using supervised learning," in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2014, pp. 1789–1798.
[30] P. E. Utgoff, "Incremental induction of decision trees," *Mach. Learn.*, vol. 4, no. 2, pp. 161–186, 1989.
[31] J. Yli-Huumo, D. Ko, S. Choi, S. Park, and K. Smolander, "Where is current research on blockchain technology—A systematic review," *PLoS ONE*, vol. 11, no. 10, 2016, Art. no. e0163477.
[32] P. Zheng, Z. Zheng, J. Wu, and H.-N. Dai, "XBlock-ETH: Extracting and exploring blockchain data from ethereum," *IEEE Open J. Comput. Soc.*, early access, May 6, 2020, doi: 10.1109/OJCS.2020.2990458.

RADOSŁAW MICHALSKI is currently an Assistant Professor with the Department of Computational Intelligence, Wrocław University of Science and Technology, Wrocław, Poland. His research interests include social influence, diffusion processes in complex networks, and machine learning. He has coauthored over 50 publications in these areas, ten of which were published in influential journals. Since 2018, he has led the Blockchain Exploration Research Group (BERG), which is focused on analyzing blockchain as a complex network. In 2019, the group developed an open source platform called Dru, which facilitates the research on blockchain–this tool has been used to gather data for the research presented in this manuscript.

DARIA DZIUBAŁTOWSKA received the Engineering degree in applied mathematics from the Wroclaw University of Science and Technology, in 2019. She is currently pursuing the master's degree with the Wroclaw Higher School of Applied Informatics. She has been a member of the Blockchain Exploration Research Group, since 2018.

PIOTR MACEK received the Engineering degree in computer science and the master's degree in data science from the Wroclaw University of Science and Technology, Wroclaw, in 2018 and 2019, respectively. He is currently working as a Software Engineer and focuses on data processing. Since 2018, he has been a member of the Blockchain Exploration Research Group, which is focused on analyzing blockchain and complex networks.